SIGNAL PROCESSING FOR BIOLOGICALLY-INSPIRED GRADIENT SOURCE LOCALIZATION AND DNA SEQUENCE ANALYSIS

A Dissertation
Presented to
The Academic Faculty

By

Gail L. Rosen

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
in
Electrical and Computer Engineering

School of Electrical and Computer Engineering
Georgia Institute of Technology
August 2006

SIGNAL PROCESSING FOR BIOLOGICALLY-INSPIRED GRADIENT SOURCE LOCALIZATION AND DNA SEQUENCE ANALYSIS

Approved by:

Dr. Paul E. Hasler, Committee Chair
Assc. Professor, School of ECE
Georgia Institute of Technology

Dr. David V. Anderson
Assc. Professor, School of ECE
Georgia Institute of Technology

Dr. James H. McClellan
Professor, School of ECE
Georgia Institute of Technology

Dr. Mark T. Smith
Professor, Skola för informations- och kommunikationsteknik (School of Information and Communication Technology)
Kungliga Tekniska högskolan (Royal Institute of Technology, Stockholm)

Dr. Oliver Brand
Assc. Professor, School of ECE
Georgia Institute of Technology

Date Approved: June 12, 2006

Dedicated to my mother, my father, and Diamantino.

Thank you for your love, support, and encouragement.

ACKNOWLEDGMENTS

The Ph.D. process is a journey. In a long journey, one needs the best guidance, and I am grateful that Paul Hasler was my guide. I would like to thank Paul, my advisor, for making this thesis work possible. I admire his great heart and enjoyment of challenges. I have learned so much from him – most of all persistence and patience. I appreciate his continuing encouragement and guidance and helping me to shoot for the stars. I would like to thank the rest of my dissertation committee including David Anderson, Oliver Brand, Jim McClellan, and Mark T. Smith. Their review and comments on my thesis have been invaluable.

I would also like to thank many people who I have met along the way who have been of immense help here at Georgia Tech. I would like to thank Jeffrey Moore for imparting rudimentary coding theory knowledge and helping me embark on studying DNA in that light. At a point in our careers, where sinking or swimming seemed equiprobable, we were able to soar with our combined effort. It was also great that we both bonded through eccentric tastes in electronic music while our simulations were running, adding laughter to make it through stressful days. I would like to thank Raviv Raich for teaching me the rigors of mathematic notation and LaTeX. His rigor and thoroughness in everything he does is very inspiring. I would also like to thank Mark T. Smith at KTH Stockholm for helping to show me "just how easy" implementation can be and the confidence to solder. I would also like to thank Jiri Janata and Ryan Cantor in the Chemistry department for providing the turbulent plume data. I would also like to thank David Hertling for his patience and clever solutions in dealing with logistical problems; he is a gifted coordinator.

I would also like to thank Ronald Schafer for being a kind ear and inspiration. Also, I would like to thank Mark J. T. Smith at Purdue for founding the CSIP Student Action Committee which I became quite active on for a while and really helps make CSIP a better place. I would like to thank Clyde Lettsome for being an inspiration, confidant, and a good

friend; his collective calm in the face of all challenges infinitely inspires me. A thanks goes out to Nikolaos Vasiloglou for his faith in me, and it meant a lot to me that NVasil Laboratories wanted to fund our DNA work. I would even like to thank the major obstacles on my way to the PhD, as I learned from from them, and they made me stronger.

I would like to thank all the friends and kind people I met at Georgia Tech. I will forget some, but I will list them anyway: Kulsoom Abdullah, Farshid Delgosha, Adriane Durey, Nick Gastaud, John Glotzbach, Heather Hopper, Sam Li, Brett Matthews, Apurva and Anna Mody, Robert Morris, Maneli Noorkami, Genevieve Raich, Mina Sartipi, Greg Slabaugh, Martin Tobias, and Rajbabu Velmurugan. I would also like to thank the ICElab: Arindam Basu, Brian Degnan, Chris Duffy, Jenny Fan, Ethan Farquhar, Christal Gordon, Dave Graham, Jordan Gray, QBear Levi, Erhan Ozalevli, Kofi Odame, Thomas Peng, Ryan Robucci, Venkatesh Srinivasan, and Chris Twigg. Also, I thank Janet Myrick, Christy Ellis, and Carla Zachery for making the GT bureaucracy smoother. I would like to thank Suzette Willingham's kindness which eased the hard times and brings smiles to the world.

I would like to extend a big thanks to the CETL STEP program, which offered me a chance to enhance my teaching skills and test them out in the real-world. I would greatly like to thank Donna Llewellyn and Marion Usselman for their efforts in making the program possible and flexible. I would like to thank Tammy McCoy and Kamau Bobb who were a pleasure to work with every week. I would like to thank all the teachers at Tri-Cities High School, especially Mr. Coker and Mr. Inman, who let me introduce "crazy" signal processing labs into their Trig/Pre-Calc classes. Also, I owe invaluable thanks to David Terraso for publicizing this work at Tri-Cities.

I would like to thank AT&T Research Laboratories and the National Science Foundation for funding most of my way. The many great people and mentors I met at AT&T such as Jim Johnston (JJ), Charles Thompson, Schuyler Quackenbush, and Patricia Wirth. All of whom are great researchers with great hearts. My summer interning at AT&T was unforgettable, working with a fun and creative boss, JJ, on audio technology, which will

always be my pasttime. JJ taught me how research should be conducted - a little passion and creativity with lots of hard work.

Also, at AT&T, I will never forget ”having-to-share” my office with an intern that summer, who almost could not find his desk under my ”stuff” when he arrived. He did not seem phased by the chaos then or now, and I thank Diamantino Caseiro for bringing order to my life with his steadfast love and encouragement; he inspires me and gives me strength everyday.

I would also like to thank Chris Cramer for helping to improve my english even further and Ed Byrnes for posting links that helped my research. I would like to thank #ewokvillage for keeping me sane by keeping me appraised of the world while I’m hidden away in my engineering corner.

This thesis would not have been possible without my mother. One of my earliest memories is of her reading to me from the book called The Brain, and driving my mind wild with scientific curiosity. She, herself, was always a technology buff, being the first one (or ”the only one” as my dad would say) to get the newest consumer electronics on the block, such as the VCR. The turning point, was in 1983, when she bought a complete Commodore 64 system (including disk drive!) for the family. My life was changed forever as I taught myself programming, and the possibilities seemed endless (and they are with the C64!) But beyond science and technology, she gave me an inquisitiveness for life, the drive to achieve, and unconditional love and encouragement. I would also like to thank my father who gave me the motivation to overcome any obstacle, and my Aunt Helen who was very supportive.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Biological signal processing can help us gain knowledge about biological complexity, as well as using this knowledge to engineer better systems. Three areas are identified as critical to understanding biology: 1) understanding DNA, 2) examining the overall biological function and 3) evaluating these systems in environmental (ie: turbulent) conditions.

DNA is investigated for coding structure and redundancy, and a new tandem repeat region, an indicator of a neurodegenerative disease, is discovered. The linear algebraic framework can be used for further analysis and techniques. The work illustrates how signal processing is a tool to reverse engineer biological systems, and how our better understanding of biology can improve engineering designs.

Then, the way a single-cell mobilizes in response to a chemical gradient, known as chemotaxis, is examined. Inspiration from receptor clustering in chemotaxis combined with a Hebbian learning method is shown to improve a gradient-source (chemical/thermal) localization algorithm. The algorithm is implemented, and its performance is evaluated in diffusive and turbulent environments. We then show that sensor cross-correlation can be used in solving chemical localization in difficult turbulent scenarios. This leads into future techniques which can be designed for gradient source tracking. These techniques pave the way for use of biologically-inspired sensor networks in chemical localization.

CHAPTER 1

BIOLOGICAL SIGNAL PROCESSING

In the 1950's, speech signal processing was born out of the need to automate speech analysis and synthesis. Models of the vocal tract were paralleled to electrical transmission lines, making the estimation of speech parameters possible through linear methods. At the same time, Watson and Crick discovered DNA as the code to life and described its structure. It was inevitable for DNA to be paralleled to computational circuitry.

First, scientists deciphered the genetic code, the way DNA maps three nucleotides to an amino acid which is the intermediary step to making a protein. In the past few decades, biologists are slowly deciphering genomic sequences and with improved electronics, whole genomes can now be mapped quickly. Only in recent years has the human genome project spurred the engineering community to solve biological problems such as gene discovery and protein interaction pathways.

DNA sequence analysis is an interesting yet limited field since the sequence only tells us the code and not function. Environmental factors and biological pathways resulting from this code have just as much weight as their DNA predecessor, resulting in a which-is-more-important the chicken-or-the-egg argument. A good review of the move from DNA to whole function study can be found in [30]. Functional genomics encompasses many challenging problems, and future work will shift to this field as the fine details of genomic sequences are completed.

One of the first processes well-understood through functional genomics is chemotaxis. Chemotaxis is the mechanism by which a cell senses and responds directionally to a chemical gradient; for example, an amoeba tracks its food in Fig. 1.1(a). It has been shown that organisms use spatial sensing mechanisms to compare receptor stimulation among different parts of the organism and then move accordingly [26]. Also, it has been observed that a cell's receptors begin to cluster towards the gradient direction when the gradient is

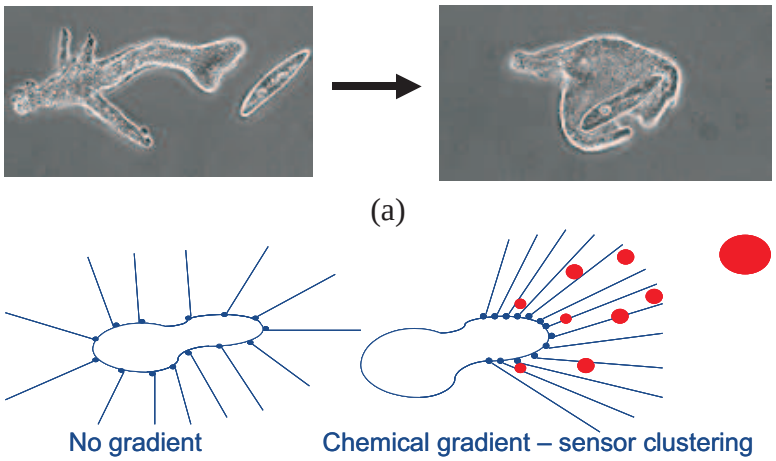


Figure 1.1. a) An amoeba locates food in a chemical gradient with mobile receptors on its membrane. It is able to gain better directional sensing of its food by clustering its receptors towards the direction of highest concentration. b) In equilibrium, the receptors are uniformly distributed around the cell. When a food source is present, the receptors exhibit a clustering behavior.

suddenly reversed [19]. We conjecture that this is due to the fact that the organism wants to increase selectivity, or its beam pattern, in that direction (see Fig. 1.1(b)).

Signal processing can be applied to a variety of biological problems. We investigate three areas: 1) analyzing DNA for coding structure and periodicity, 2) using inspiration from the way organisms mobilize in a chemical gradient field to improve odor source localization, and 3) analyzing turbulence for plume localization.

This research contributes to our understanding of bio-informatics as well as improves practical chemical localization. A parallel can be drawn between biological systems and our proposed localization system (see Fig. 1.2). Biological systems and engineered systems both follow the same architecture of instructions (DNA) $\rightarrow$ implementation (protein pathways) $\rightarrow$ system performance (bio-function).

We show how signal processing techniques can be used to solve problems on both the DNA level and the functional level. First, we develop and show how linear algebraic techniques can be used to analyze DNA. When these linear techniques are used for a strict conditions, that of universal error-correction in the sequence, they are not fool-proof, but more

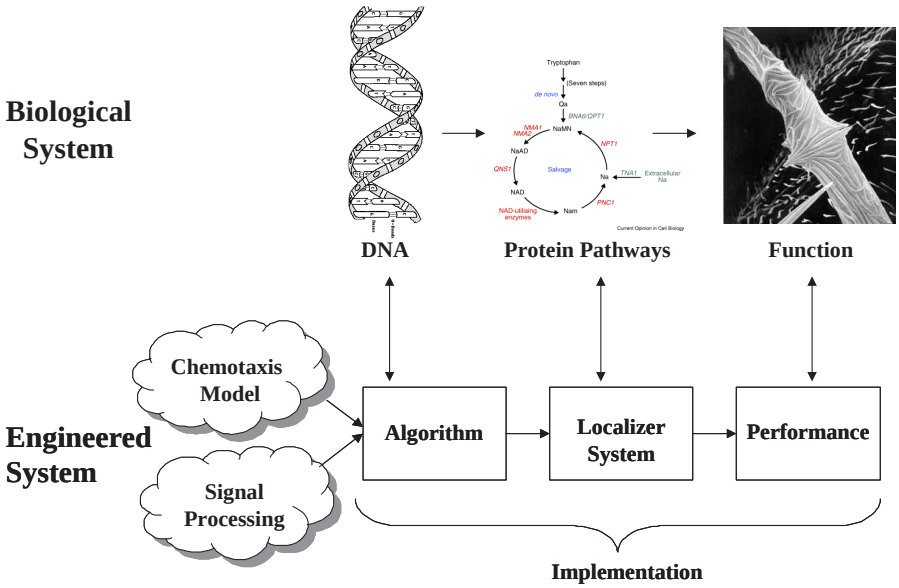


Figure 1.2. Paralleling an engineered system to a biological system.

general linear methods are superior in detecting imperfect periodicities (approximate tandem repeats), a classically difficult problem. Secondly, we show that signal processing can be used in conjunction with biology to improve an engineered system, in this case odor localization. We show that chemoreceptor clustering improves sensor array performance, and so there is potential for improvement as the chemotaxis model complexity increases. Also, we analyze turbulent plume experimental data and show that a cross-correlation method, such as interaural time delay in binaural hearing, improves turbulent plume localization. In addition, the overall research supplies concrete examples of how signal processing techniques can be used to analyze biology and how biology can help us engineer better systems.

1.1 Organization of the Dissertation

In the rest of this document, we will present our work in analyzing DNA with signal processing techniques, engineering a better chemical localizer using biological inspiration, and assessing and localizing turbulent plumes. In Chapter 2, we examine the structure and redundancies found in the DNA sequence and how errors/mutations occur. In Section 2.2, we review previous work in studying structure and redundancy in DNA and propose that DNA

be analyzed within a Galois finite-field framework due to its symbolic nature. Using this framework, we present a subspace partitioning method in Section 2.3 to detect a general linear block error-correcting code in a DNA sequence. Then, in Section 2.4, a linear dependence (LD) test is developed to detect dependence between frames of data; the conditions for this test are less strict than the subspace partitioning method. Due to its generality, the LD test is more robust to mutations and finds approximate tandem repeats very well.

In Chapter 3, we review chemical detection and tracking and biology's role in solving these types of problems. In Chapter 4, we review previous chemotaxis-based approaches based on single and multi-nodes and develop a multi-sensor single-node chemotaxis-inspired algorithm to track diffusive sources. While the others either use a random walk or gradient-difference strategy, our algorithm utilizes a localized chemoreceptor clustering phenomenon to enhance performance. In Chapter 5, we analyze experimental turbulent plume data for its spectral properties and correlation then explore modeling the plume as a filter. Using short-time correlations from small events, or parcels, in the plume, we are able to localize its angle-of-arrival in Section 5.5. In Chapter 6, we summarize the contributions of the work and suggest topics for future research.

CHAPTER 2

STRUCTURE AND REDUNDANCY IN DNA

The genetic code is the instruction on the translation of nucleotides to amino acids which later form proteins, but this example is only one of many signals encoded in DNA. It is well-known that these protein-coding regions have the lowest mutation rates in the DNA strand. So, the question arises: how does DNA protect itself from error? A thorough review of DNA signal content, redundancy, and mutational mechanisms is presented. With knowledge of frameshift and substitution errors caused by mutations, methods to detect a linear coding structure and approximate tandem repeats are developed.

Since the introduction of the Watson-Crick model of DNA, scientists have been trying to make sense of the long sequence, millions of bases long for simple organisms and billions long for higher ones, composed of four nucleotides. The genetic code, the mapping of nucleotide triplets (codons) to amino acids or protein-coding, was one of the first discoveries and indicated that signals encoded in DNA can be paralleled to digital signals. Discovered when the invention of the computer was in its infancy, the progressive theory would have seemed far-fetched by biologists without overwhelming proof of its existence. Then, inspired by Shannon's mathematical theory of communication [92], scientists began to explain DNA within an information theoretic framework (e.g. [32]). After 30 years, many functions and signals in DNA still remain unknown, and scientists have conjectured that non-protein-coding regions (97% of human DNA) is unused junk [42]. Recent studies reveal that binding sites and initiation signals exist in these non-coding areas, and mutation errors in these regions cause diseases [94]. Huen [42] and Stambuk [97] show that properties of non-coding regions contain a finite amount of algorithmic content. Discovering the signals and function in these areas is just the beginning of genome discovery. In the first section of this chapter, DNA structure, composition, mutation, and repair are reviewed to give a biological perspective on DNA's inherent redundancy and repair mechanisms. Then

an overview of mathematical representations of nucleotides is given, and DNA complexity is examined. The last two sections present new approaches to DNA redundancy analysis: a method for finding block error-correction codes [86] and a linear dependence test to search for approximate tandem repeats [85].

2.1 DNA Structure and Error

2.1.1 DNA Composition and Repeats

DNA is composed of four bases or nucleotides, A (adenine), G (guanine), T (thymine), and C (cytosine). A and G are considered purines (R), and C and T are considered pyramidines (Y) with purines being the larger of the two. This size imbalance between creates an affinity between purines and pyramidines, and stability is only reached when A bonds to T (2 weak hydrogen bonds) and C bonds to G (3 weak hydrogen bonds). An information-theoretic view of the bonding affinity is explored in [59] where each bond type corresponds to a binary value as well as the R/Y status. Exploring each molecule possible with a parity-code representation, it is found that the most stable and optimal molecules are surprisingly all even parity ones that correspond to natures choice [59]. Nucleotides are strung together using a rigid backbone composed of sugars and phosphates, and there are two backbones on either side with a staircase of nucleotides in the middle. Because of the bonding constraints, nucleotides on each side have a complementary pairing (e.g. $A-T$ and $C-G$). The weak complementary bonds make DNA easy to unzip in replication but can also make it susceptible to interfering molecules; thus for protection in its stable state, the double strand curls into a helix. It has been found that certain nucleotide repeats help DNA to wrap into the curved state. The dinucleotides, AA and TT, are placed at certain phases from each other and cause an average periodicity of 10.55 ± 0.01 in the DNA sequence; AG and CT, still a purine-purine/pyramidine-pyramidine adjacent pair, also aide to the helical twist [101].

Many other nucleotide patterns are built-in due to DNA's physical structure. Packing DNA into its chromosomal shape influences base-periodicity. To illustrate, in some

genomes, the helix coils around circular beads, histones, to make a nucleosome. Each nucleosome is wrapped in 200 base pairs of DNA, and there is shown to be a 200-base (and stronger multiples) periodicity introduced by this structure [101]. Also, Trifonov reports a 10.5 base periodicity introduced by the coiling around the histone, and a 10-base repeat of *TATAAACGCC* has a high correlation to the nucleosome regions [109].

Strings of nucleosomes are packed to make two chromatids; the chromatids are then joined to make a chromosome. The connection piece in the center of a chromosome is called the centromere, and it too is constructed by nucleotide patterns. Highly repetitive regions, containing *TGGAA* repeats, compose the sequence and are believed to aid to the stability of the connection/replication-aide [109]. A more fascinating nucleotide series are the telomeres, the ends of the chromosomes used to buffer genes from the environment. When DNA replicates, the process shortens DNA on each iteration with each cell division (human DNA shortens 50 base pairs, bp, per replication); to prevent nucleotide loss from eventually truncating a gene, telomerase elongates the ends with repetitive sequences such as *TTAGGG*, sometimes for thousands of bases [110]. As we age, telomerase expression weakens, genes no longer have protection from being cropped, and cells die. On the contrary, when telomerase is over-expressed, cells tend to live much longer and divide more frequently, resulting in cancer [36]. 90% of tumoric growths exhibit excessive amounts of telomerase! Sequence periodicity and repeats play a vital role in stability of the overall structure.

Lets delve deeper into the nucleotide organization. Various DNA regions are correlated to specific functions or signals, and a famous function is that of protein-coding or the coined genetic code. First DNA must be transcribed to RNA; in this process, an initiator protein binds to a promoter site on the DNA region, detects a start signal, reads triplet nucleotides (or codons), detects a stop signal, and synthesizes an RNA sequence. Ribosomes then bind to the RNA sequence, and for every triplet segment traversed, an amino acid is sequentially attached to the ribosome, making a chain of them. When the translation process is finished,

the amino acid chain folds and makes a protein. Equivalent amino acid chains can fold into different proteins, thus there is no one-to-one mapping from amino acid sequence to protein type which makes protein-prediction a challenge. The interaction of proteins then form more complicated processes, and this is when life as we know it, comes about. DNA is just the recipe for how to make proteins, but genes (the parts which translate amino acids or what is known as "coding) actually rely recursively on the proteins to initiate their translation. A good review of the complex interactions between DNA and its cellular environment and a "chicken or the egg debate about which came first, proteins or DNA, can be found in [29]. The protein coding described also correlates to periodicities and patterns seen in DNA. A 3-base period is widely observed and studied in the literature [101], [7], [99]. This is due to an imbalance/bias of bases: temperature-dependence combined with the fact that a purine is more likely to be found in the first position and a pyramidine in the last position of a codon [101]. To facilitate the translation process, signals are embedded in the DNA sequence to create binding sites for transcription factors. This corresponds to a $TATA$ box, one or more $TATA$ repetitions, in prokaryotes (cells without a nucleus), and a Pribnow box, $TATAAT$ repeats, in eukaryotes (cells with a nucleus). Also when DNA is transcribed to mRNA, a Shine-Delgarno sequence, $AGGAGG$, is passed on which acts as the ribosomal binding site. These identified patterns and sites already give seemingly random DNA a clear deterministic structure. Schneider presents a comprehensive list of DNA signals recognizable by pattern and analysis of their information content [88].

DNA structure diverges in the two different cell type, prokaryotes and eukaryotes. One of the distinct differences is the fact that prokaryotes have short DNA length (100 to 1000 times less than eukaryotes), and protein-coding sequences consist of contiguous strings of nucleotides. Eukaryotes, on the other hand, contain exons, the protein-coding parts, interleaved with introns, non-coding intermissions in genes. The introns are spliced out when the gene is transcribed to mRNA, and their purpose is still relatively unknown. Also, eukaryotic DNA has more repetitions, such as microsatellite regions, than prokaryotes [20].

Do introns enhance the fidelity of genes? Do an abundance of repeats lead to better error-protection? Many questions still remain.

2.1.2 Mutations and the Replication Process

While trying to view DNA from a computational standpoint, it is difficult to keep a 3-D, biological perspective. Scientists give a rough error-rate Fig.corresponding to 10^{-10} mutations/nucleotides when DNA is copied. So, what are these mutations and how can we quantify them? Mutations mostly occur due to 1) accidental bonding of Brownian-motioned biological elements to DNA, or 2) electromagnetic radiation providing enough energy to break bonds in the structure. As an example of 1), one of the most common mutations is the hydrolysis of C to T, known as cytosine deamination. Water molecules do not have as easy access to nucleotides when they are protected in the stable helical structure as they do when DNA is unzipped for replication. In fact, cytosine deamination is 100 times more likely in replication [21]. Temperature, geometry, and environment are key factors in studying DNA mutation rates.

In addition to errors caused by clumsy molecules bumping into DNA, the copying mechanism (DNA replication) itself can introduce errors which appear structured. For example, microsatellite regions, an excess of repetitive sequences, are caused by replication slippage [67]; the replication polymerase slips and copies one or more extra repeats of an already duplicated sequence. Microsatellites in human DNA are associated with 14 neurodegenerative genetic disorders found in [94]. Having 5 to 37 repeats of CTG near the DMPK gene may have no effect while 50-80 of them could lead to male-pattern baldness, and over 80 repeats cause myotonic dystrophy, muscle atrophy caused by the inability of muscles to relax [30]. Discussed in the previous section, repeats from telomerase slippage causes increased cell division and highly correlates with malignant cancer growth.

The replication procedure alone has an error rate of 10^{-3} to 10^{-5} [21]. But DNA has an internal proofreading mechanism. When copied, the helical structure unzips and forks into two separate strands; complementary bases then attach themselves to complete the

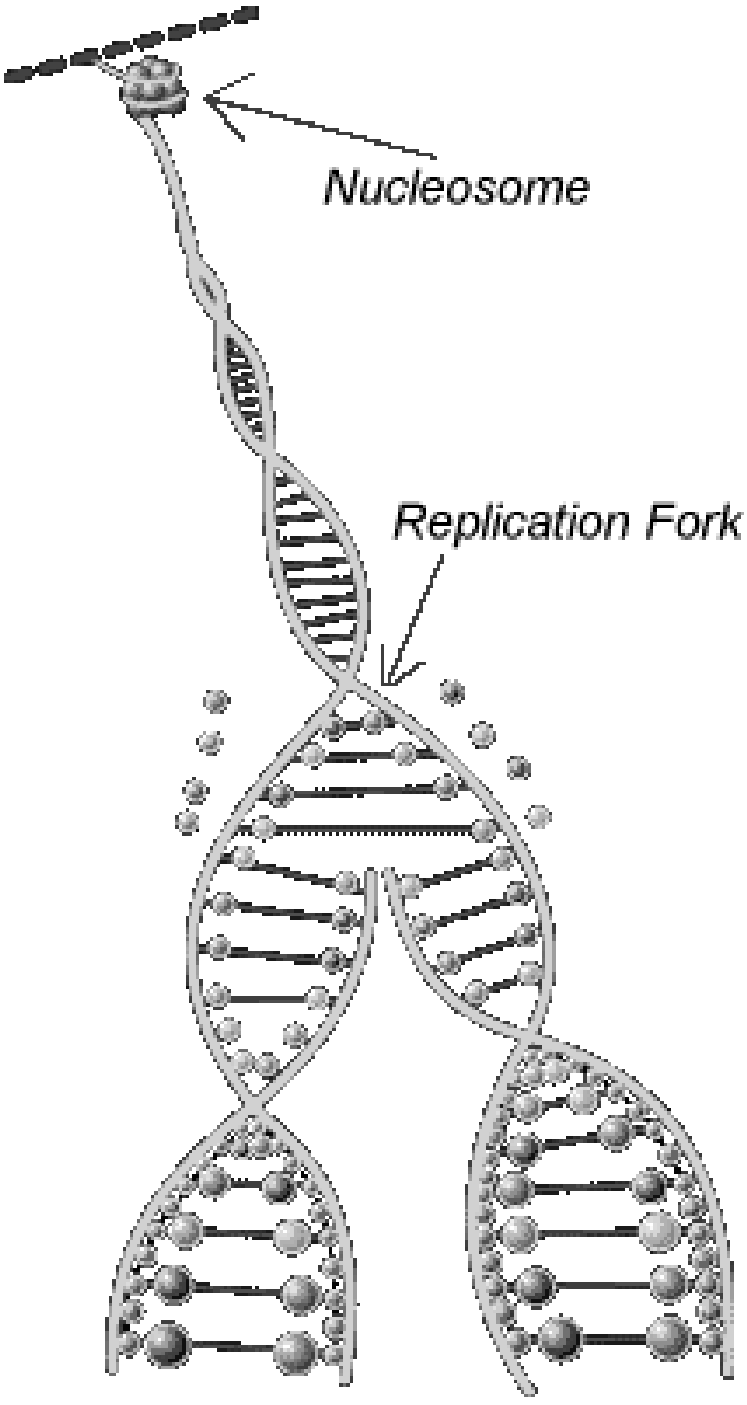


Figure 2.1. Illustration of the DNA replication fork.

new ladders (see Fig. 2.1). When a substitution error occurs, usually a purine replaces a purine ($C \rightarrow T$) or a pyramidine replaces a pyramidine ($A \rightarrow G$) in the complementary attachment, a kink develops due to the mismatch. If an unstable mismatch is detected, the

polymerase does not add more bases until the correct nucleotide is restored. This simple proofreading reduces the error rate to approximately 10^{-10} [21]. In addition to proofreading, other post replication repairs can occur. For example, ultraviolet radiation can cause damage to the DNA helix, and in nucleotide excision repair, the whole injured fragment is removed and replaced. Can understanding these repair pathways lead to better error-correcting technologies?

Also, does the speed of replication offer any insight into the complexity of the mechanism? In prokaryotes, DNA is copied at approximately 1000 base pairs/second while in eukaryotes, its around 50 base pairs/second; replication has been shown to be 100 to 10,000 times more accurate in eukaryotes. It has been theorized that fast replication is linked to lack of introns [55]. If introns slow the replication rate, is there complexity introduced by them or an encoding mechanism not found in prokaryotes?

2.2 DNA Analysis and Information Theory

2.2.1 Nucleotide Representation

Whenever one attempts to tie mathematical theory to the genome, the most important assumption is the representation of the set of nucleotides, A, T, C, G. It has even been contemplated why nature chose such an alphabet [59]. There are many representations proposed and are adapted to the type of analysis. Assessing the purine/pyrimidine structure, one can represent the purines (A and G) and the pyrimidines (C and T) with a binary representation. In addition, for four bases, one can choose a simple representation such as $A = 0$, $G = 1$, $C = 2$, $T = 3$ and use modulo operations, but this implies a structure on the nucleotides such that $T > A$ and $C > G$. For a model of the translation process, Anastassiou defines a complex representation to the nucleotides $A = 1 + j$, $T = 1 - j$, $G = -1 + j$, and $C = -1 - j$ [7]. The geometric interpretation of this representation still imposes a structure such that the Euclidean distance between A and C is greater than the distance between A and T, yet for the nucleotide quantization to amino acids, it was useful. The complex representation has been paralleled to a telecommunication transmission quadrature phase shift

Table 2.1. Table of DNA mathematical representations found in the literature. An example sequence, GCATT, its complement, and a characteristic property is given for each representation.

Example sequence: G C A T T | | | | | A A T G C	Representation	Sequence	Complement	Property
Simple Integer Assignment	$A = 0, G = 1,$ $C = 2, T = 3$	1 2 0 3 3	0 0 3 1 2	Modulo Operations
Complex Assignment (QPSK)	$A = 1 + j, G = -1 + j,$ $C = -1 - j, T = 1 - j$	$-1 + j, -1 - j,$ $1 + j, 1 - j, 1 - j$	$1 + j, 1 + j, 1 - j,$ $-1 + j, -1 - j$	Reverse and conjugate to get complement
PAM Representation	$A = -1.5, G = -0.5,$ $C = 0.5, T = 1.5$	$-0.5, 0.5, -1.5,$ 1.5, 1.5	$-1.5, -1.5, 0.5$ 1.5, -0.5	Reverse and negate to get complement
Binary Indicator Sequence	$S_i[n] = 1$ where $S[n] = i$ $S_i[n] = 0$ where $S[n] \neq i$	A: 0 0 1 0 0 G: 1 0 0 0 0 C: 0 1 0 0 0 T: 0 0 0 1 1	A: 1 1 0 0 0 G: 0 0 0 1 0 C: 0 0 0 0 1 T: 0 0 1 0 0	4-dimensional representation
Galois Field Assignment	$A = 0, C = 1,$ $T = 2, G = 3$	3 1 0 2 2	0 0 2 3 1	Symbolic Galois Field operations

keying (QPSK) constellation and used for autocorrelation analysis in [22]. Chakravarthy also uses a discrete numerical sequence symmetric about the y-axis, which can be paralleled to a pulse amplitude modulation (PAM) scheme and which preserves DNA's reverse complementary properties [22]. Also, indicator sequences (binary sequences representing the locations of the nucleotide elements) produce a four-dimensional representation yielding an efficient representation for spectral analysis [7] [99]. When modeling processes in E. Coli mRNA, a fifth base, Inosine, can be taken into account [66]. All of the representations including the one proposed for our analysis can be seen in Table 2.1.

Symbolic statistical techniques, using Markov models to represent the various nucleotide states, have been developed to predict gene sequences [18]. But a representation is needed which allows deterministic mathematical operations on a finite set of elements. Abstract algebra offers three ideas for DNA analysis: groups, rings, and fields. For a short synopsis, a group is a set of elements on which a binary (usually additive) operation has been defined, a ring can have multiplicative and additive operations but an inverse may not exist (i.e.: subtraction is possible but not division), and a field has both operations and

Table 2.2. Exponential root representation, polynomial representation, numerical label, and nucleotide label for the $GF(4)$ representation.

$$\begin{array}{rcccc} \alpha^0 = 1 & \Leftrightarrow & 1 & \Leftrightarrow & \mathrm{C} \\ \alpha^1 = \alpha & \Leftrightarrow & 2 & \Leftrightarrow & \mathrm{T} \\ \alpha^2 = \alpha + 1 & \Leftrightarrow & 3 & \Leftrightarrow & \mathrm{G} \\ 0 = 0 & \Leftrightarrow & 0 & \Leftrightarrow & \mathrm{A} \end{array}$$

Table 2.3. Addition and multiplication tables in $GF(4)$.

+	0	1	2	3
0	0	1	2	3
1	1	0	3	2
2	2	3	0	1
3	3	2	1	0

*	0	1	2	3
0	0	0	0	0
1	0	1	2	3
2	0	2	3	1
3	0	3	1	2

their inverses [108]. If one wishes to have a wide range of operations available for linear algebraic analysis of a set of elements, a finite field is the preferred framework.

DNA is a symbolic set and in no way can be characterized as a group, ring, or field. After all, the commutivity and associativity of DNA is unknown. As a result, we look solely upon the fact that if linear algebra were available as a tool, ease of analysis awaits. Therefore, using this logic only, we choose to analyze DNA as a finite field and will inspect the results to assess the validity of this framework.

We propose a mapping of nucleotides to a Galois field [108] of four, $GF(4)$. Since $GF(4)$ is an extension field of $GF(2)$ (any $GF(2)$ binary pair corresponds to one of four $GF(4)$ symbols) , we can create labels (see Table (2.2)) for the nucleotide elements with $GF(2)$'s primitive polynomial:

$$\alpha^2 + \alpha + 1 = 0 \tag{2.1}$$

This places on DNA the following Galois field properties: the elements are commutative under addition and both commutative and associative under multiplication as well as having an identity element, additive, and multiplicative inverses. This abstraction of elements to integer labels makes finite field theory an attractive framework.

The polynomial in (2.1) can be manipulated in addition, multiplication, and its inverses in $GF(4)$. Refer to [108] for a detailed derivation. For reference, we show the resulting

operation tables in Table 2.3.

A question arises when using these operations in a linear space. What does it mean for a vector of nucleotides to be self-orthogonal? In $GF(4)$, the inner-product of [2 1 0 3] with itself is 0. Abstraction of pure mathematics to a physical system introduces anomalies, and its implications are not obvious. We will use this framework to analyze redundancy in DNA and draw conclusions about the advantages and disadvantages of a finite-field framework. A more in-depth discussion is in the Galois Field and DNA section at the end.

2.2.2 Complexity Analysis And Information Theory's Role

Since DNA nucleotides are a finite alphabet of four, the strand lends itself to being viewed as information storage. Naturally, information theory can be used to analyze the sequences. We review information theoretic studies, coding theory models, and then finally comment on tandem repeat (periodic repeats) redundancy in the sequence.

2.2.2.1 Information Theoretic Studies

The pioneers of information theory applied to DNA were directly inspired by Shannons theory of communication [92]. Gatlin developed entropy and divergence measures to quantify complexity in DNA. By investigating measures of entropy, we can look at basic measures of information content. The entropy, or information capacity of a sequence, is maximized when all four nucleotides are equiprobable:

$$H = \sum_i p_i log_2(p_i) = \frac{1}{4}\sum_{i=1}^{4} log_2(\frac{1}{4}) = 2bits \tag{2.2}$$

In many species, the bases are not equiprobable, but temperature dependent. Three bonds exist in C and G bases while only two exist in A and T. Thus, it takes more energy to make C and G, and it has been found that GC content is higher in warm-environment organisms than cold-environment. For example, Micrococcus Lysodeikticus [32] has the following base frequencies: $Pr(C) = Pr(G) = 0.355$ and $Pr(A) = Pr(T) = 0.145$. Therefore, the entropy for this organism utilizing the first part of (2.2), is 1.87 bits, which implies

redundancy from this imbalance. For the example data given later in the subspace partitioning method, a segment from the E. coli K-12 $MG1655$ coding region sequence has the following composition: $N(A)/N = 0.262$, $N(C)/N = 0.281$, $N(T)/N = 0.206$, $N(G)/N = 0.25$ where $N(X)$ is the number of X nucleotides and N is the total number of nucleotides in the sequence. Therefore, the sequences entropy is at a near maximum with 1.99 bits.

A simple entropy measure like (2.2) indicates if a nucleotide bias exists in a sequence. Since then, new measures have been developed such as entropic profiles of various-length genomic sequences and computing the Kullback-Leibler distance between the histograms of these profiles [78]. Kortokov introduces a non-parametric decomposition method which analyzes the mutual information between experimental and artificial sequences; the method is effective in revealing latent periodicities [52].

Beyond entropy measures, other methods have been developed to examine bias and statistical properties of DNA. An in-depth study on dinucleotide bias yields that CT, GC, AG, and TA pairs are found more frequently than other pairs [23]. In [23], a novel cumulated and unwrapped phase technique is used to measure $+90^\circ$ transitions corresponding to dinucleotide pairs; the importance of the open-reading frame orientation (a window of triplet-coding DNA and the direction in which its read) is reinforced by the results: a clear accumulation of positive transitions is found when in the correct orientation. It is shown that codons (triplet nucleotides) have a clear preference for purines in the first position and a pyramidines in the last position [9]. In [9], autocorrelation analysis shows that frame 0 in gene regions is partial to being composed of one of 22 fixed trinucleotide sequences (a number close to the 20 codons expected). Another analysis sheds an interesting perspective on triplet-coding bias by viewing it as noise. Using a state function combined with a distortion correction measure, the bias in the triplet coding areas is compensated (the sequence is whitened) [90]. Schneider illustrates DNA nucleotide bias for each position through an easy-to-read sequence logo graph [89]. In [97], through Cantor set theory, it is shown that non-coding DNA segments possess characteristics of natural languages, while coded DNA

sequences have a more deterministic structure. Techniques to study information content and bias begin to quantify DNA's implicit structure.

2.2.2.2 Coding Models of DNA

In [12], Battail hypothesizes a nested coding scheme for DNA error-correction and how introns may be involved; nested coding would preserve ancient, essential "supergenes while evolutionarily transforming them to new genes as well. Battail also addresses the important point that without mutation, evolution cannot occur so there is fine balance between preserving DNA information and change. He also calls for a more in-depth study of error-correcting structure in DNA. The supergene concept is further backed by Arques who finds the same trinucleotide frequencies, positioned in frame 0, common to both prokaryotic and eukaryotic protein coding regions [9]. Battail also recently conjectures that DNA may have resulted from a series of evolving repeats commonly found in introns and presents a framework for replication decoding [13]. Only a partial knowledge of the coding constraints is needed to decode a message in the "multiple unfaithful repetition model; this property makes the model attractive since little is known about the DNA encoding structure.

Much work has been done by May et al. to study E. coli translation initiation sequences using block and convolutional coding models [65] [64] [63] [106] [74] [87]. mRNA is viewed as a noisy encoded signal and the ribosome, which translates the sequence, as the decoder. Several biological and chemical factors are used to parameterize the ribosomal decoding model, and the performance is quantified by distance of the received sequence to the signal motif. The block code model is effective in recognizing the ribosomal binding site, and the convolutional model easily distinguishes between translated and un-translated sequences. Various convolutional code generators are being investigated to improve recognition of the binding site [74]. Also, a graph theoretic approach using Hamming-distance based coding spheres clearly exhibits distinctions between valid E. coli initiation signals and random sequences [87].

Lastly, Benos seeks a simple, deterministic recognition code in transcription regulation,

but a probabilistic model based on the approach yields better results [15]. A two-way model is computed which associates how a protein binds with a series of nucleotide sequence, and vice versa, how a nucleotide sequence binds with a series of proteins. Affinity, an energy measure, and specificity, a preference measure, is determined for each signal region for use in subsequent prediction.

Since DNA is a finite, symbolic sequence, the use of coding theory to analyze these sequences is a natural extension. Examinations of nucleotide bias and signal recognition are explored under this framework.

2.2.2.3 Tandem Repeat Detection Notes

Another lengthy article could be written on just techniques of tandem repeat detection. Here we will give a short overview of the problem. From structural studies, we know organisms, especially eukaryotic, DNA have repetitive regions. The repeats can be classified into three categories: SSR: Simple Sequence Repeat (ex: *AT AT AT*), VLTR: Variable Length Tandem Repeat (*CATG CACATG CATGTG*), and MPTR: Multi-period Tandem Repeat (*CAG CAT TAG CAT CAG CAT TAG*) [38]. There have been various techniques to classify these [88] [38] [105] [51] [16] [53] [104]. Most algorithms have complex heuristic, combinatorial, or dynamic programming approaches. In [88], a periodicity transform is used to plot several periodic/near-periodic regions vs. position on one simple graph. It is one of the most flexible (by using different detection thresholds) and efficient (periodicities vs. nucleotide position) representations, but only base substitution mutations (not frame offset mutations) are taken into account.

2.3 Determination of an Underlying Linear Code

Liebovitch presents the first search for an error-correction code in DNA where a single parity-bit search is developed [57]. He presents the novel idea that there might be more to DNA repair than just a polymerase detecting irregular kinks in the sequence. Is there a structure inherent in DNA based upon classical information theory which helps it maintain

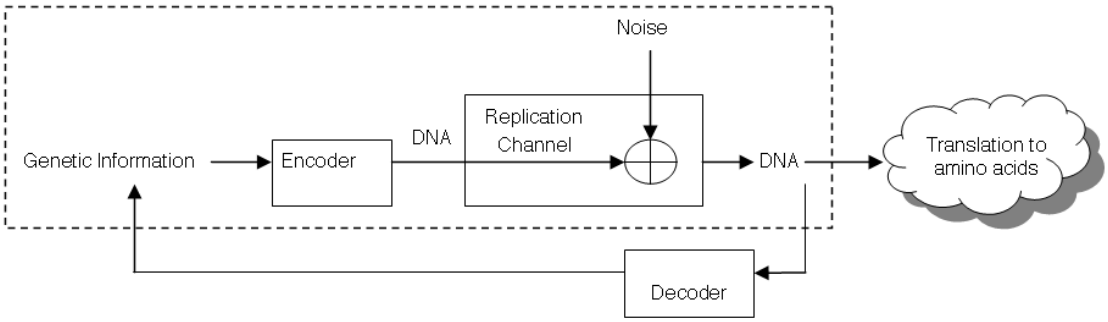


Figure 2.2. Our noisy channel model of genome replication with underlying coding assumption.

high fidelity? While his methodical investigation does not reveal the presence of a consistent single parity-bit code, the experiment provides inspiration for future investigations and context for the complexity of the problem. Thus, there is a need for a general approach, which we explore in this section, to discern an (n,k) block coding structure from DNA sequence content.

2.3.1 Modeling the Replication Channel

Communication channel models can be paralleled to DNA processes. In one doctrine, the channel is assumed to be the amino acid translation from nucleotide triplets [32]. In May et al., the channel is the actual replication process [66]. The latter is good for mutation modeling since transcription and copying of DNA is a noisy process. "Proof-reading mechanisms are observed during DNA replication, and when the activity of these polymerase mechanisms are blocked, error rates increase from 10^{-10} to 10^{-3} [21]. We use a model similar to Mays since errors occur directly on the DNA strand in replication while errors in the translation process can also occur in the formation of amino acids and proteins. In our framework, DNA is the medium in which genetic information is transmitted from generation to generation.

In Fig. 2.2, we assume that the DNA is the sequenced genomic data available in GenBank [3] and that our goal is to examine the dashed-line encompassed area and uncover the encoder scheme; in other words, we wish to infer structure from the noisy output to

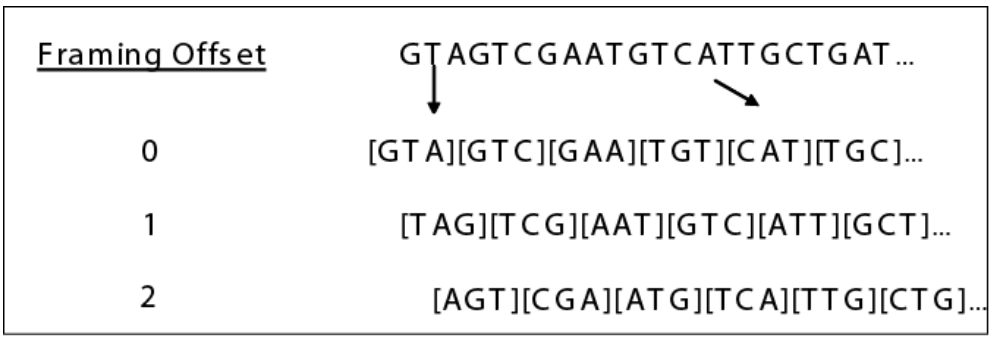


Figure 2.3. Illustration of vector framing for n=3.

retrieve the original genetic information. Also, if our assumption is correct and DNA is encoded in a linearly redundant fashion, our analysis will uncover it. In this system, we know nothing about the encoder nor the original information, thus, system identification and deconvolution methods cannot be used. We will assume that the encoder is linear and try to characterize it given such output.

2.3.2 Subspace Partitioning For (n, k) Codes

In this investigation, our primary goal is to identify and characterize any linear constraints that might appear in regions of a sequence. Lacking the benefit or prior knowledge regarding the location, duration, or dimensionality of subspace partitioning in the sequence, we propose a method that generates a complete orthogonal basis set oriented to a local region of data. The basis set is used to decompose the sequence (equivalent to a coordinate transformation). The consistent presence of nulls in the transformed sequence indicates both the presence and the dimension of linear subspace partitioning in the data.

The first assumption is a fixed codeword length, n. The DNA elements are grouped into a matrix, $\mathbf{V} = [v_1 v_2 ... v_N]$ where the length of the entire DNA sequence is and v_i is length n. The alignment of the frames relative to the starting point will be referred to as the framing offset. A choice of a particular framing offset will be referred to as the frameset, or open reading frames as coined in the biological literature. Given the frame length n, there are n unique framesets. See Fig. 2.3 for illustration of all frameset groupings.

We apply the Gram-Schmidt algorithm using finite field operations to the sequence of vectors to yield a complete set of orthogonal basis vectors, e_1, e_2, e_n. In the event that the entire sequence consists of vectors lying in a subspace of dimension less than n, we introduce random vectors and continue to iterate Gram-Schmidt until the basis set is complete. This yields a transform matrix $\mathbf{G}$ that is clearly full-rank, as it consists of n orthonormal vectors.

Once an orthogonal basis is formed from the first j frames of data, the v_is for $i > j$ are decomposed into components of each of the basis vectors. This is simply a coordinate transformation and can be described by:

$$t_i = \mathbf{G}v_i \text{ where } \mathbf{G} = [e_1, e_2, \ldots e_n]^T \tag{2.3}$$

Provided that the data has been framed correctly when applying the Gram-Schmidt algorithm, a linear coding redundancy can be detected by noting consistent null coordinates over a region in the transformed sequence of length-n vectors, $t_1, t_2, , t_{N-j}$. This null detection would indicate whether a subspace of the actual n-dimensional space exists.

Knowing nothing of the dimension or alignment of the data, we must apply the algorithm for many codeword lengths. For a given codeword length and for a given locality in the sequence, we apply the algorithm n times to account for each framing offset. For each of the n iterations, the vector frameset is offset by one element of the sequence to guarantee that if length-n codewords are present in the sequence, one of the framesets will be properly aligned.

Subspace Partitioning Algorithm Outline

1. *Obtain the orthonormal basis, $e1, e2, , en$, by Gram-Schmidt orthogonalization of j number of v_i frames where $j \geq n$. Form the transform matrix, $\mathbf{G}$, from this set.*

2. *Decompose the sequence into its basis components, $t1, t2, tN - j$, across all possible framing offsets.*

3. *Note the persistence of nulls in t_is. Calculate confidence by comparing against the probability of sequential sets of randomly chosen vectors having the same subspace partitioning.*

It should be noted that on finite fields, non-zero element vectors can have an inner product of zero (the additive identity element of the field), thus self-orthogonal vectors can exist. The situation sometimes arises in which a subspace is characterized entirely by self-orthogonal basis vectors. For this reason, the coordinates in the transformed vector sequence associated with these self-orthogonal basis vectors are always zero. In this case the decomposition cannot proceed and the algorithm must be terminated, reframed, and started anew.

Given the copious volume of data produced by iterating the algorithm over numerous frame shifts and codeword lengths, a visualization method is devised to aid in the search for consistent subspace partitioning. For each frameset, consistent nulls in the decomposed vectors are noted in an attempt to characterize the unoccupied subspace. A null-subspace indicator vector is used to mark the locations of nulls found consistently in the data. Each shift in sequence results in an update of the indicator vector. If the vector remains unchanged across iterations, a probabilistically-based value increases to indicate confidence in the presence of subspace partitioning (as the probability of randomly-chosen vectors possessing the observed subspace partitioning diminishes). We can then plot the confidence as a function of sequence index *i* across all possible framing offsets.

2.3.3 Results Of The Subspace Partitioning (SP) Method

The algorithm is capable of detecting and characterizing linear subspace partitioning in any sequence provided that such structure is manifest in the data. For a given sequence, all such structure can be found provided that the algorithm is run for every possible framing offset and for every possible codeword length.

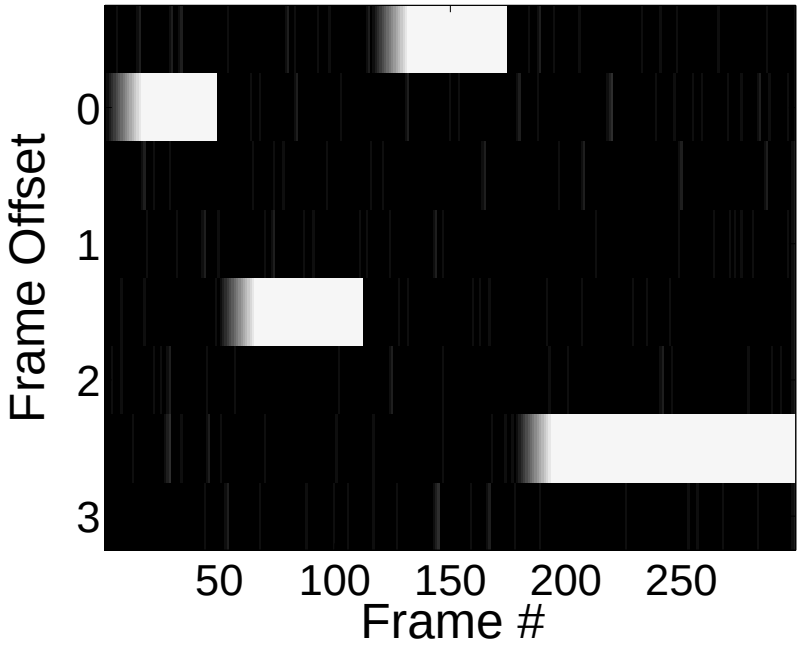


Figure 2.4. Linear subspace partitioning results for a $(8, 5)$ **coding test data.**

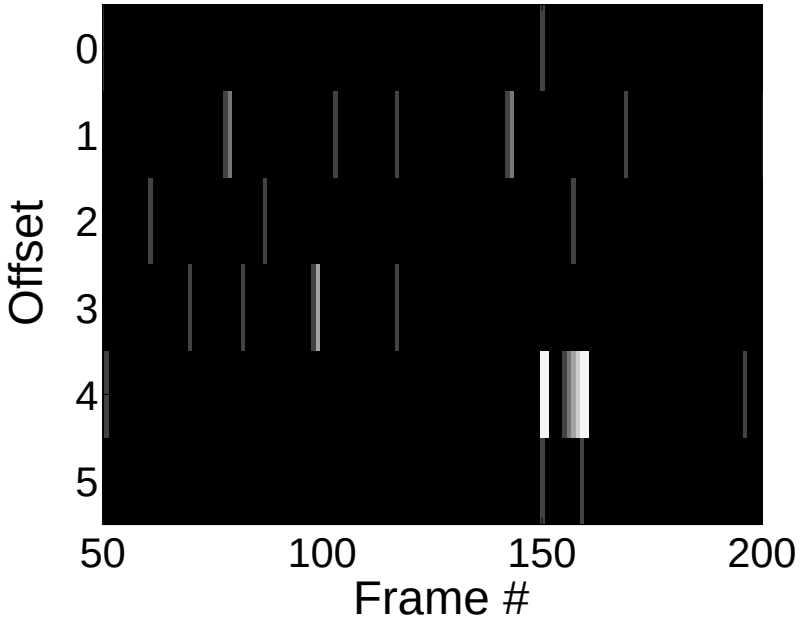


Figure 2.5. Linear subspace partitioning results for a subsection of an $n = 6$ **E. Coli K-12** $MG1655$ **sequence.**

By way of illustration, a test sequence is generated to occupy a five-dimensional subspace of an eight-dimensional vector space. This constitutes an $(8, 5)$ linear block code in $GF(4)$. Running the algorithm on this sequence for $n = 8$ yields the confidence image

shown in Fig. 2.4. Interstitial symbols are introduced throughout the sequence to illustrate the robustness of the algorithm to framing offsets. When framing offsets are introduced in the sequence, the region of high subspace partitioning confidence simply migrates to the corresponding row in the diagram.

These confidence stripes by themselves say nothing of the dimensional occupancy of the underlying sequence. Rather, they are used as search tools to simplify the analysis of large volumes of data. Their presence alerts us to the location of subspace partitioning in the sequence, at which point we can retrieve the local indicator vector to observe that, indeed, there are three dimensional nulls present throughout the duration of each of the confidence stripes.

The linear subspace partitioning algorithm is then tested using an E. Coli K-12 *MG*1655 sequence (GenBank accession code *NC*_000913). The result is shown in Fig. 2.5. A consistent linear block code is not observed to be present throughout the whole sequence, but some regions are oriented in the same subspace for several consecutive frames, denoted by the aggregated intensity of the light bars. Other sequences and fragments, prokaryotic and eukaryotic, were tested and yield similar results of an intermittent subspace.

The subspace partitioning algorithm requires two conditions from the sequence. Firstly, the algorithm uses nulls in a transform to indicate subspace partitioning. This requires that the coordinate system described by the transform be properly oriented. The transform matrix is guaranteed to be properly aligned for exactly one of the possible framesets, provided that the structure in question is present from the outset of the sequence. If there is an onset of structure in the data at a later point in the sequence, it may not be found. This stems from the primacy effect inherent to the Gram-Schmidt algorithm: the coordinate system (basis set) produced is oriented according to the order in which vectors are presented.

Secondly, the component decomposition algorithm is defeated by the case in which the Gram-Schmidt algorithm produces a fractional basis set. This is because finite field arithmetic allows for the existence of self-orthogonal vectors. The situation sometimes

arises in which Gram-Schmidt produces a coordinate subspace whose complement contains entirely self-orthogonal vectors. While this situation is rare (7 out of 75 times in processing the E Coli DNA strand), it is impossible to perform the decomposition discussed here when it does occur. In this way, it creates "blind spots" for the algorithm: certain combinations of codeword length and framing offsets are self-orthogonal and cannot be analyzed using Gram-Schmidt.

The subspace partitioning method is an adaptable algorithm for general redundancy analysis. It identifies and parameterizes dimensional occupancy in a region independent of framing, provided that the structure is present from the outset. This algorithm can be more generally applied to any sequence for which it is suspected that coding properties are present. The algorithm could readily be adapted in a classification scheme for data of unknown origin or for cryptographic/cryptanalysis tasks in which the code or encryption scheme is unknown.

2.4 Linear Redundancy and Tandem Repeat Detection

Now with a nucleotide representation and field-defined arithmetic operations, we can extend the linear algebraic techniques used in the previous section. As reviewed in the first section, DNA is shown to be highly redundant. To analyze redundancy, we develop a method, the linear dependence test, to search for localized regions of linear dependence in sequence data. The linear dependence (LD) test indicates the mere existence of a subspace while the subspace partitioning method from the previous section tells us the subspace's orientation. If we can determine that a subspace exists and is present for a portion of the data, we can use this as a starting point for further examination of its orientation (as explored in [86]). The LD test determines local redundant regions and is useful as starting point for further complexity analysis such as detection of tandem repeats.

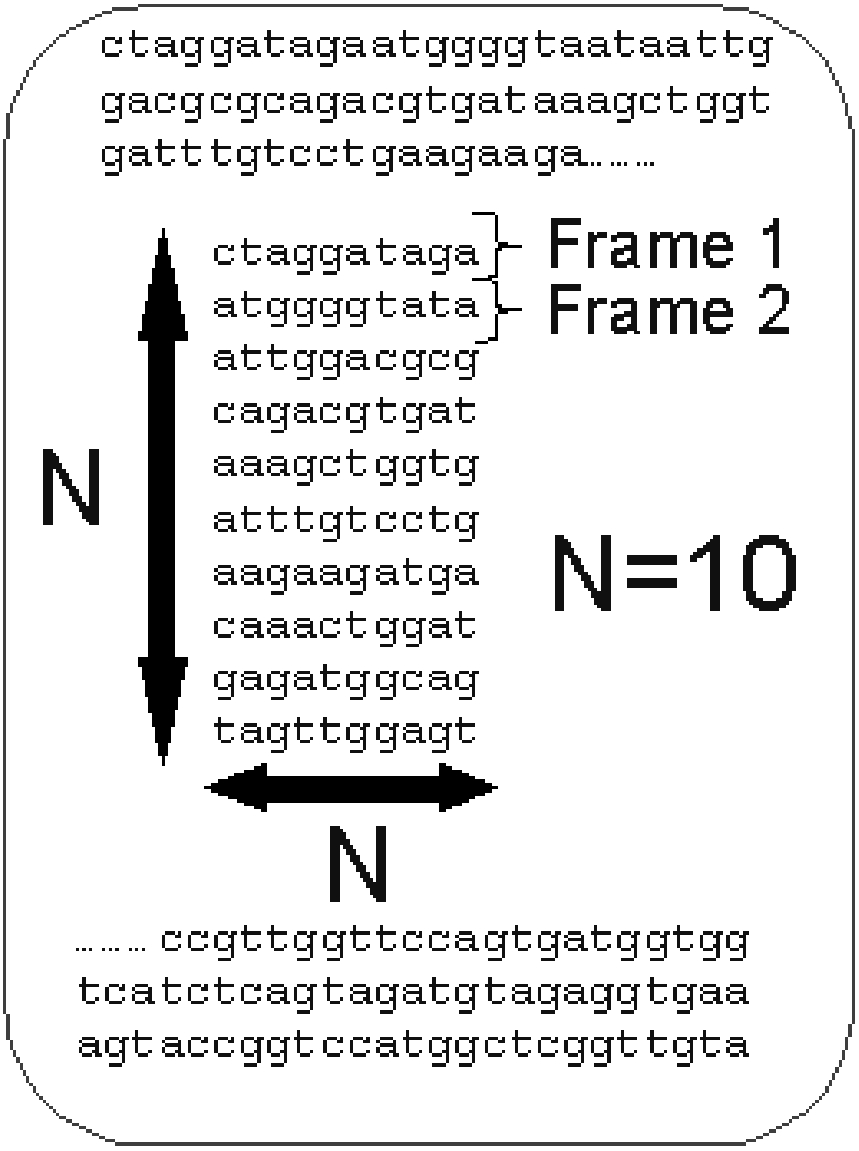


Figure 2.6. Illustration of $N \times N$ windowing for the LD test, where $N = 10$.

2.4.1 Linear Dependence Test

In the LD method, an N^2-length window of the data is reshaped as an $N \times N$ matrix as shown in Fig. 2.6. This matrix occupies a maximum of N-dimensions. In the linear dependence test, the rank of each $N \times N$ window is computed to find its dimensional occupancy; the rank computation is based on a recursive Gaussian-elimination [34] modified for $GF(4)$ arithmetic. Then the data is incremented by an N-length frame each time, thereby creating a slowly moving $N \times N$ window which moves by N nucleotides each iteration until the

entire sequence has been traversed. A weight, I, is increased linearly, $I = I + 1$, on each iteration if rank-deficiency is found in consecutive windows segments. The outline of the LD technique:

1. *For analysis frame length, N, collect N consecutive vectors to form NtimesN window.*
2. *Perform a rank computation of the $N \times N$ matrix.*
3. *Increment by one frame for each iteration.*
4. *Note consistent rank-deficiency by linearly increasing I. By itself, this method is a measure of linear dependence in regions of the data but not necessarily globally as needed for a block coding scheme. For global linear coding to be present, the basis vectors would have to form a consistent subspace over all frames in a sequence. Thus, this algorithm is more localized and detects approximate and even slowly varying rank-deficient regions.*

2.4.2 Sequence Data Source

Using the online Genbank database [3], we select the Yeast Chromosome I sequence (accession code: NC_001133) and a human satellite region (accession code: $HSVDJSAT$) as our test data to illustrate the algorithm.

2.4.3 Linear Dependence Test Results

In graphs of Figs. 2.7,2.9,2.10, and 2.11, the x-axis corresponds to the frame number in which the $N \times N$ window begins, and the y-axis denotes our algorithm running for all $N - 1$ frame offsets needed to test all possible groupings (see Fig. 2.3 for illustration) of the data. If an insertion or deletion occurs and effectively shifts a redundant portion forward or backward by one or two bases, the rank-deficient portion will still be shown but in another frame offset since all frameshifts are examined. If an $N - 1$ rank subspace is found, it is denoted in a dark gray, and the lower the rank of the subspace (up to $N - 4$ for the examples), the brighter the intensity; also, the higher the linear dependence persistence

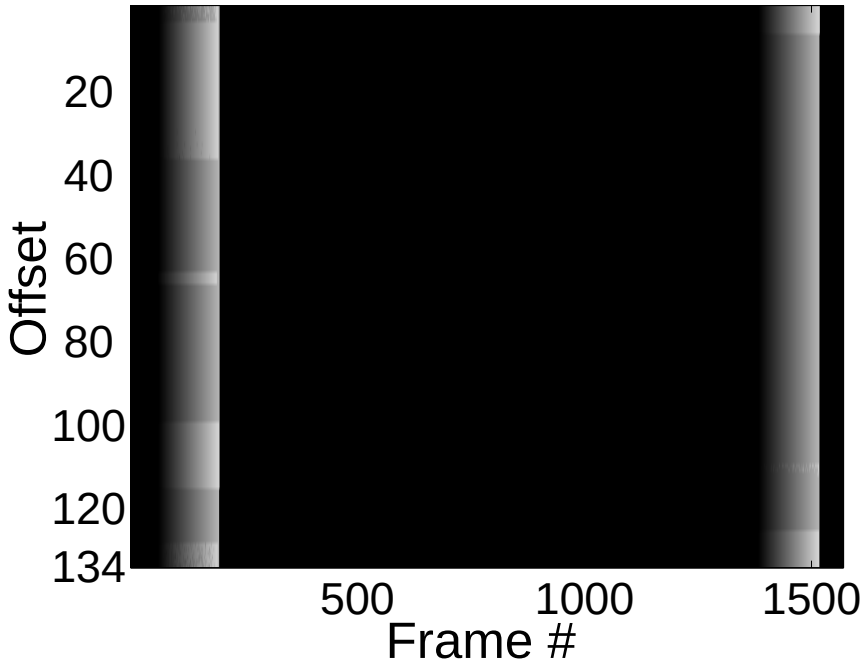


Figure 2.7. $N = 135$ **LD test for the Yeast Chromosome I sequence,** NC_001133**. Intensity increases with the length and level of the rank-deficiency. Two regions associated with the** $FLO9$ **gene are shown to be highly repetitive with the LD Test.**

indicator, I, (mentioned in the LD method's final step) the brighter the shading intensity. Therefore, the brightness of the graph is a function of two factors: the strength and length of the redundant region.

First, the algorithm was run on the Yeast Chromosome I sequence which can be seen in Fig. 2.7. In Fig. 2.7, two notable redundant regions of over 17000 bases are found to have rank-deficiency for $N = 135$. Even though data is only deficient by one or two dimensions, visually inspecting a portion of the data in Fig. 2.8 shows the frames are almost identical to each other, indicating that a tandem repeat is present.

In [38], it is found that the $HSVDJSAT$ sequence, a repetitive satellite region of 1985 bases in the human genome, has a tandem repeat of 19 bases from 1195-1553. Using the LD test, one can easily see the tandem repeat in Fig. 2.9. While the strong repeat is from 1150 - 1728, a longer redundant region starting around base 900 is detected by using an offset of 6. As a reference for comparison, this sequence was run for $N = 20$, and the redundancy is weak when compared to the $N = 19$ graph. Therefore, the LD test can easily

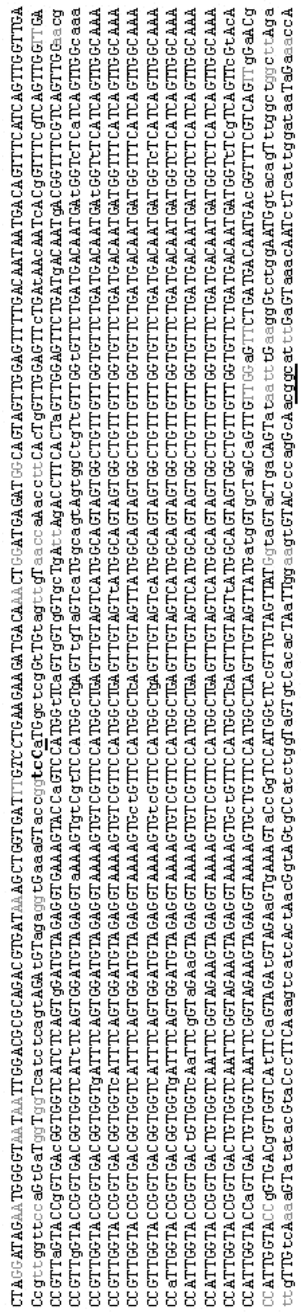


Figure 2.8. Annotation of a near tandem repeat of 2025 nucleotides (bases 25310 ⟶ 27335 **in** *NC*_001133**) in Yeasts chromosome I. Underline denotes an insertion from the previous frame, Bold denotes a region retained after/around a deletion occurring from the previous frame,** *Italicize* **denotes a region before a deletion, UPPER CASE denotes conserved portions, lower case denotes substitution errors/ sequence differences, and light gray denotes portions where multiple base substitutions occur for a particular base.**

distinguish between strong and weak periodic regions as well as find a longer periodicity by testing all frameshifts.

Exhaustively running the algorithm for more N, it is also found that a strong periodicity

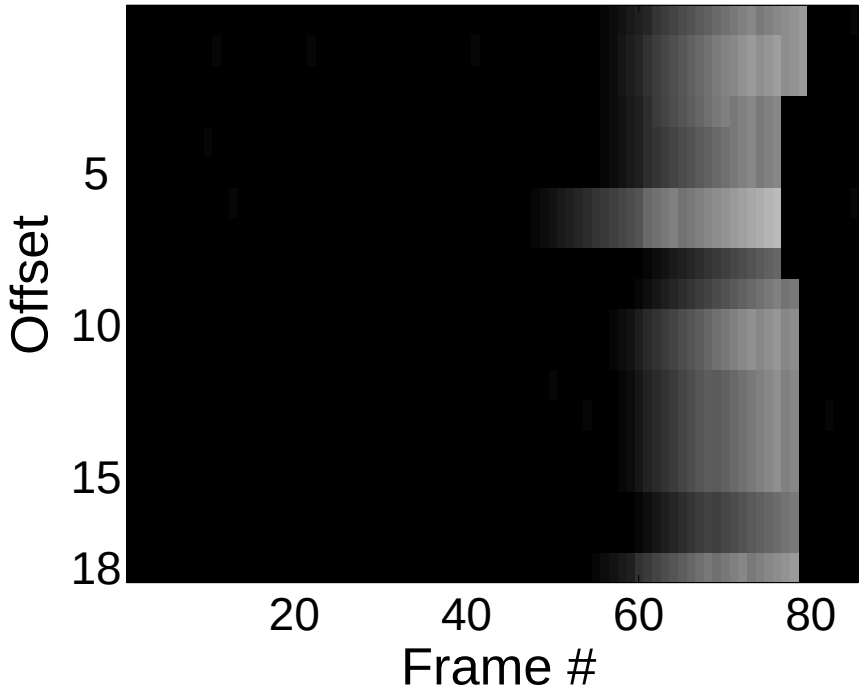


Figure 2.9. $N = 19$ **LD test for a human satellite sequence,** $HSVDJSAT$**. Intensity increases with length and level of rank-deficiency. At offset 6, a 893 base region exhibits a 19 base repeat.**

of 24 bases exists as seen in Fig. 2.11. At an offset of 12, there is a periodic region of over 1100 bases, which is longer than the periodicity found in the $N = 19$ runs. Hauth has recently reported a periodicity of 48 from 1190-1553 [39], but with most tandem repeat algorithms, the $N = 24$ periodicity or multiples is difficult to find. The *mreps*2.5 algorithm [51] did not yield a periodicity or multiples of 24 for this sequence. This may be due to the fact that no exact repetition exists. In Fig. 2.12, a portion of the $HSVDJSAT$ region is shown, and no two frames are equivalent because of mutational errors. For current tandem repeat algorithms, this is a problem because they are based on exact frequencies, but our algorithm detects redundancies and therefore can easily identify near-periodic regions. In Fig. 2.12, the lower case and light gray nucleotides show regions where the nucleotides may have mutated to other nucleotides (known as substitution errors) which occur when DNA is replicated. The light gray ones are interesting because they represent substitution of one or more nucleotides, usually dinucleotides, which is a surprisingly often occurrence. The italicized and bold symbols indicate before/after regions where a deletion has occurred. The

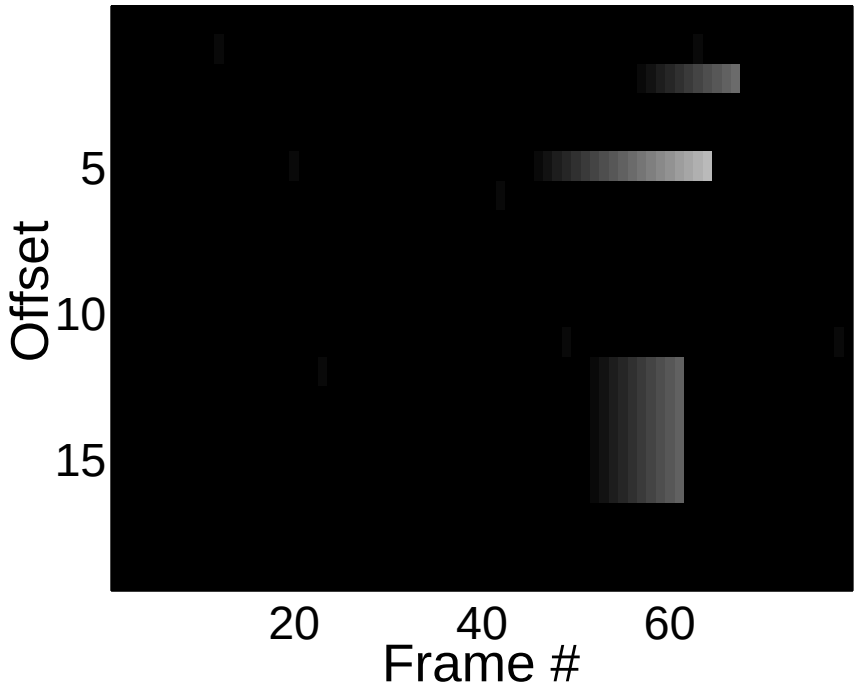


Figure 2.10. $N = 20$ **LD test for** $HSVDJSAT$**. Compared to the** $N = 19$ **case, redundancy is weak.**

underlined nucleotides represent an insertion from the previous frame. Finally, to illustrate frame to frame conservation, upper case nucleotides are used.

The LD algorithm does not search for exact repeats or matching patterns, instead, the rank-deficiency of the nucleotide window indicates similar structures between the vectors. Despite these errors which throw other algorithms astray, the LD algorithm easily detected the periodicity of 24.

2.4.4 Discussion: Galois Field And DNA

There are a few disadvantages to using $GF(4)$ in conjunction with linear algebra. The clearest is the self-orthogonal vector property. For this to occur, the number of nucleotides in the frame must be an even number; if we look at Table 2.3, a number added to itself is 0, and this can occur for an inner-product of an even-element symmetric nucleotide vector, $ATAT$ or [0 2 0 2]. Before more complex linear algebraic operations can be used, this anomaly must be dealt with.

The proposed analysis in this chapter yields a quick way to visually inspect periodicity

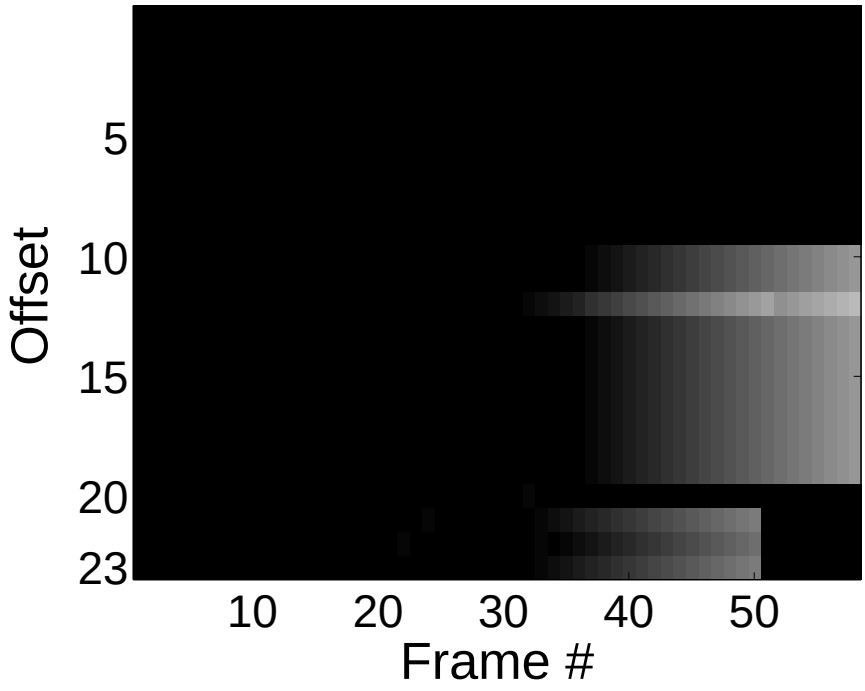


Figure 2.11. $N = 24$ **LD test for** $HSVDJSAT$**. At offset 12, a 1200 base region exhibits a 24 base redundancy. This is longer than the** $N = 19$ **case, and is prone to many mutational errors which makes it hard to find.**

in DNA regions. We now address the validity of the finite-field framework. From Fig. 2.12, it can be seen that the algorithm is robust amidst various errors. This is due to the fact that the finite field structure preserves the symbolic nature of the nucleotides. Similar structures are linear combinations of each other (e.g. $GAGA = GTGT + ATAT$). If a pure repeat exists, the dimensional occupancy of an analysis window is 1. With added insertions and deletions, the window rank increases but not enough for all basis vectors to be linearly independent of each other, thus still detecting a strong redundancy among errors. Although not all analysis can be performed because of the self-orthogonal property induced by $GF(4)$, the Galois field allows complex operations on a finite symbolic set and enables powerful tools for DNA analysis.

2.4.5 Conclusions

The subspace partitioning method is based on the hypothesis that there is an underlying coding structure in DNA used for error recovery in replication, but preliminary results

Figure 2.12. Annotation of an $N = 48$ *HSVDJSAT* region (bases 1141 $\longrightarrow$ 1976). The annotation scheme used in Fig. 2.8 is used here. An approximate repeat can be seen among insertion and deletion errors.

from our investigations do not indicate a universal coding structure. We assume this structure would occur in both protein-coding and non-coding regions. (There has been great effort in distinguishing between these gene and junk regions [18] [102].) On the contrary, mutation rates vary from region to region in the genome, and these areas may need separate treatment. Nature relies on mutations and uses errors for diversity, and it has been shown that non-coding regions (which compose over 97% of the human genome) are more susceptible to mutation than protein-coding regions. Also, frequency of mutation can vary from one gene to another; different genes in corn showed variation of mutation rates by 400-fold [21]. While a universal error-correction code in whole genomes is not probable, DNA could be encoded with varying schemes from region to region.

In the linear dependence test, we develop an algorithm which finds near-periodic DNA regions, common to genetic disorders, in a fast iterative process. In addition, it is shown that using a finite-field framework enables the use of linear algebra's massive toolbox. Two sequences are analyzed via the LD algorithm, and expected tandem repeats are found in each. An unexpected approximate repeat of 24 bases is found in the *HSVDJSAT* sequence. The

discovery is due to the algorithms ability to detect redundancy amidst abundance of mutation which other algorithms do not tolerate. The linear dependence test is a simple way to find imperfect periodicities and remains robust in substitution, deletion, and insertion errors.

Finally, there is still work to be done in investigating error-corrective properties and redundancy in DNA as well as studying its protection mechanisms. More complex DNA coding models should be taken into consideration such as convolutional coding models when testing for such structure. Also, the presented linear algebra framework can be extended to DNA computing research involving development of optimal codewords [43] as well as novel complexity studies.

CHAPTER 3

ELECTRONIC NOSES: BIOLOGICALLY-INSPIRED TECHNIQUES

Electronic noses (e-noses) have been around for approximately 20 years, but they have only recently attracted a flood of attention from engineers. For example, most landmine clearance techniques are usually slow and/or expensive, and better methods are needed. Also, recently, the demand for electronic noses has skyrocketed because of the need to detect explosive vapors and biological agents. Now, it is as important as ever to lower the cost of these devices for widespread use.

Everything has an odor signature. Humans can smell the chemical presence of volatile compounds in the air but animals, with more sensitive noses, can detect the presence of substances that appear odorless to humans. Currently, animals such as dogs and rats are still the most cost-effective odor trackers for the level of accuracy. For example, APOPO [5], a landmine removal organization, traces explosive vapor emanating from landmines, by using the extreme sensitivity of the rat's nose (see Fig. 3.1). This acute sense of smell can be attributed to the fact that rats and dogs have more chemoreceptors and more developed olfactory bulbs than humans. Although the training of rats has sped up landmine discovery, training each rat still takes a considerable amount of effort and money. Electronic vapor detectors are much needed to reduce this cost, and typical applications of this technology are alcohol breath analyzers, gas leak detectors, food quality sensors, and exhaust emission detectors.

3.1 History

For the first ten years, the main topics were **detection** with the main focus on chemical sensor development and **discrimination** using pattern recognition and classification techniques. Only in the past ten years, has the problem of chemical source **localization** arisen

Figure 3.1. APOPO International [5] trains sniffer rats to detect explosives and diagnose disease. Animals are still the best detectors and locators of chemicals.

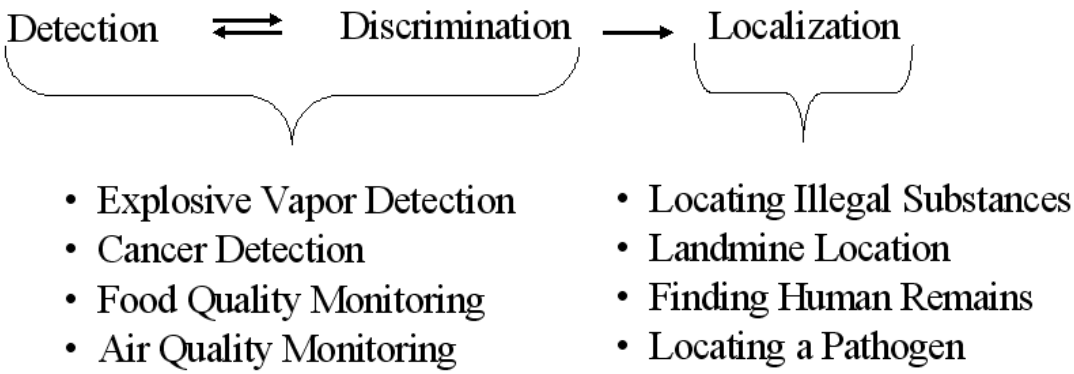


Figure 3.2. The three main areas of e-noses are inter-related. Detection and discrimination have a circular relation. To detect a level of a chemical, you must be able to discriminate it from others and to discriminate between odors, you must have a level of detection. Chemical localization is dependent on how well you discriminate the chemicals to begin with. We also list important problems and applications for each area.

which combines a variety areas, namely array processing, statistics, and robotics. Fig. 3.2 shows how these three topics are intertwined and how they can be applied.

The first step to designing an electronic nose is to develop chemical sensors. The ideal chemical sensor will measure a value that is proportional to the chemical intensity at that point. This has proved to be a challenging problem because of physical constraints, such as changes in the sensor material over time due to chemicals present resulting in measurement

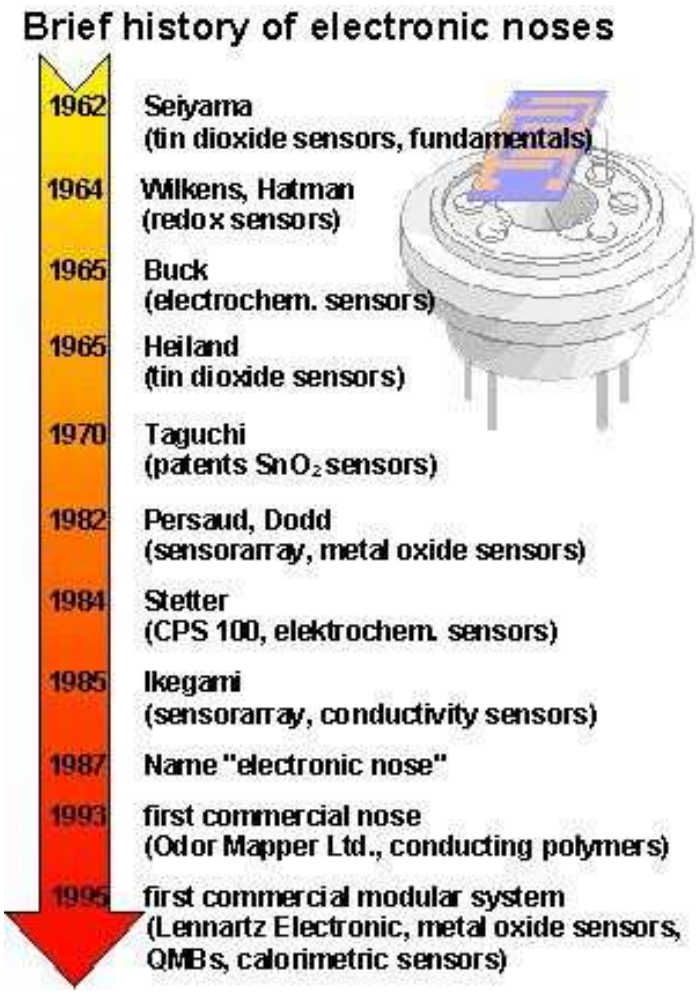


Figure 3.3. Brief History of the Electronic Nose listing the preliminary chemical sensor technologies. [2]

"drifts". The first and simplest sensors developed were the tin oxide sensors. The underlying principle of conductometric sensors (also called chemoresistors) is the conductivity change that occurs when gaseous molecules react chemically with metal oxide semiconductors or organic conducting polymers [69]. These are what are known as electrochemical sensors. Chemical sensors can be divided into several categories: thermal, mass, electrochemical, and optical. The trade-offs are similar to other sensors with sensitivity vs. specificity. A brief timeline of e-noses can be seen in Fig. 3.3.

Potentiometric chemosensors of the MOSFET (Metal-Oxide-Semiconductor Field-Effect-Transistor) types were developed to utilize a gate that is made of a gas sensitive metal as

a catalyst for gas sensing [69]. One such example are ChemFETs, chemically sensitive field-effect transistor arrays [73].

Widely used in military applications are SAW (Surface Acoustic Wave) sensors. Gravimetric odor sensors detect the effect of absorbed molecules on propagation of acoustic waves. The two main types of gravimetric sensors include QCM (Quartz Crystal Microbalance) and SAW (Surface Acoustic Wave) devices that are configured as mass change sensing devices in the electronic nose [69].

The above examples have been in use for decades, and there are many other types of sensors which we do not highlight here. Recently, breakthroughs in MEMS (Micro-Electro-Mechanical Systems) [54] and carbon nanotubes provide interesting solutions. Recently, there has also been advances in neuromorphic sensors [93] and analog vlsi interfaces [14] [8]. Analog circuits have the added value that they are low-power, and adding analog interfaces to amplify/process the chemical sensor signals puts more processing in analog circuitry while also pre-processing the signal for easier backend processing on-chip or externally.

The backend for discrimination utilizes mostly signal processing techniques have been a starting point in designing electronic noses [41]. For example, the following electronic nose discrimination algorithms use signal processing techniques:

- In [91], neural networks and black-box modeling, including ARMAX models, are used to improve recognition of biological samples.
- In [25], wavelet transforms are used to improve feature extraction from electronic nose measurements.
- In [35], pattern recognition and machine learning techniques improve electronic nose discrimination.
- In [103], a uniform sensor network estimates a chemical source's coordinates from binary observations and compares the performance of two estimators.

3.2 Biological Inspiration and GSP

Figure 3.4. The University of Arizona Neurobiology Laboratory builds robots which mimick moth behavior.

Animals are still the state-of-the-art in chemical detection. Now, many scientists and engineers are turning to biology for inspiration. Cells offer insight into localization with their directional sensing (see Fig. 3.5), and mammals offer insight into odor discrimination [79] . GSP (Genomic Signal Processing) is used to study these mechanisms by dynamically modeling biological networks. The aim of GSP is to unite the theory and methods of signal processing with biological insight. With GSP, we not only understand biology better but can then design better algorithms.

Examples of GSP-based localizers:

- Sensor cooperation techniques from chemotaxis improve an array's directionality and speed up chemical source localization in low SNR regimes. [82]
- Bacterial foraging strategies help develop optimal search methods. [72]
- One of many biomimetic systems, [58] shows an enhancement of plume tracing by mimicking moth behavior.

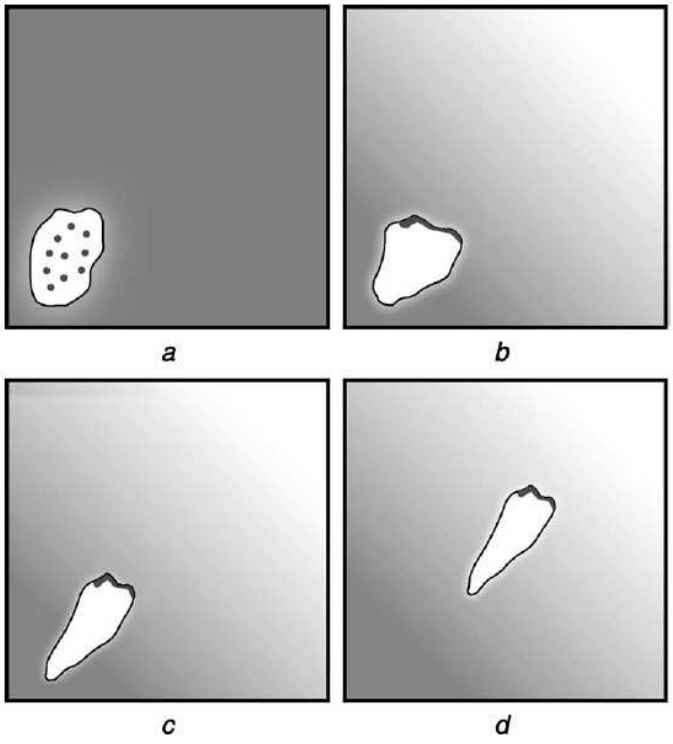


Figure 3.5. Chemotaxis is the process of how cells mobilize in a chemical gradient and is one of the most well-understood post-genomic functions. Researchers gain insight from the robust adaptation of this process to improve optimization algorithms and the localization problem. a) The cell's receptors are in equilibrium. b) In a gradient, the receptors congregate towards the side closest to the source. c) The morphology of the cell changes. d) The polarized cell migrates towards the source. [71]

- The robotics community uses swarm behavior in animals to speed up the time required to locate the odor source. [40]

3.3 The Call for Standardization of Parameters in Chemical Localization

Because of all the various techniques for chemical localization including 1) stationary, 2) single-node mobile, and 3) multi-node mobile solutions, these different scenarios make it difficult to assess the performance of a real system. For example, all too often the intensity of the source and diffusion constant is not given. Performance metrics are needed to compare these techniques across the board. For example, the localization time for a discrete-step mobile solution could be given in a step-size metric which could then translate to time depending on the speed of the vehicle. The standardization of performance

metrics for these systems could be broken down into these three areas and then further improved upon to compare cross-approaches.

For all scenarios, environmental factors are important:

1. A model of the field should always be given. If possible, the source size/intensity and diffusion constant/rate should be given, otherwise the intensity drop-off should be given, or the field should be characterized.

2. If turbulence is present, it should be characterized so that other turbulence algorithms can be prepared.

3. The sensor specificity, sensitivity, and characteristics should be noted, including regions of linearity and saturation.

Parameters necessary for the **stationary** cases:

1. The distance from sensors to the source.

2. Sensor placement.

3. Probability of estimation error.

4. Time needed to reach the estimate to within a certain probability.

Parameters necessary for the **Single and Mobile node** case:

1. Vehicle speed

2. Distance travelled

3. Turbulence introduced by self-movement.

4. Initial sensor placement

5. Restrictions on search space.

Mobile node only:

1. Resolution of node overlap.

2. Differences between node algorithms or the cooperation between nodes.

3. Best localization time, average localization time, and worst localization time for the nodes.

3.4 Major Challenges in Chemical Detection, Discrimination, and Localization

Challenges remain in the electronic nose field:

- Chemical localization in low-to-medium turbulence (e.g. common windy or convective environments)

- Chemical detection and localization in highly turbulent environments, such as HVAC systems

- Better discrimination of chemicals such as bioagents in multicomponent mixtures (e.g. using models of canine olfaction)

- Fast localization through mobile implementations

- Better localization and discrimination through sensor fusion, combining odor information with other extra-sensory information (Most animals use multi-sensory information to track down a source.)

In the past decade, many strides have been made to improve chemical sensors and their detection, discrimination, and localization [41] [75], [44] [27] [72] [62] [77]. Yet, animals are still used to locate landmines and illegal substances because e-noses are still not cheap and accurate enough to mass-produce for widespread use (e.g. the Cyranose 320 has a retail price of $8000 in 2006 [4]). Now, genomic signal processing enables us

to understand many cellular functions. This understanding will help us engineer better bio-inspired systems such as the electronic nose.

CHAPTER 4

MODIFIED HEBBIAN LEARNING IMPLEMENTATION FOR LOCALIZING AND TRACKING DIFFUSIVE SOURCES

4.1 Diffusive Source Localization

Many objects (biological) that we want to track have signals (heat or chemicals) that diffuse rapidly in their environment resulting in a very difficult tracking problem. Multi-sensor or array signal processing can improve the tracking of diffuse signals, thereby decreasing the cost of large sensor arrays [47] and mobile sensors [40], as well as greatly decreasing the convergence time to track these objects.

The problem of odor localization has been tackled across several disciplines. Biological optimization algorithms based on bacterial chemotaxis (BC), which can be viewed as single-sensor algorithms, can be considered for field navigation. They are useful for searching a surface without staying in a local minimum, and if a map of minima is kept in memory, a global minimum can be determined [68]. These algorithms can be used for navigating through a complicated field. However, single-sensor BC breaks down quickly in the presence of field noise (i.e. Brownian motion) [49]. Even though, bacterial foraging strategies help develop optimal search methods [72].

Interesting approaches come from robotics, but most papers do not quantify the performance of the system. One of many biomimetic systems, [58] shows an enhancement of plume tracing by mimicking moth behavior. A swarm robotic approach is inspired by insect behavior to localize a chemical source, and its performance criterion based on the actual time and distance taken to find the source was compared over group size, plume type, and various search algorithms [40].

On the signal processing side, a comprehensive detection and estimation theory framework has been developed for vapor source localization via a stationary sensor array [70]. It was determined that five sensors and four parameters are the minimal amount needed in

a three-dimensional space using this framework, and performance for the each parameter is shown via the Cramér-Rao bound vs. time for one noise level [70]. The framework was extended to a single moving sensor which uses a circle-and-attack strategy [75]. Applying the stationary framework to landmine detection using chemical sensors, optimal sensor placement around a landmine is determined and detection probabilities are plotted for the number of sensors and time samples [47], but it requires a huge sensor array size. Also in this study, the performance of the single moving sensor is evaluated. It has an advantage over biological algorithms because it allows real-time optimization of its trajectory, but the disadvantage is that it uses an extensive search time for the initial detection. Again, using a huge sensor array size, in [103], a uniform multi-node network estimates a chemical source's coordinates from binary observations and compares the performance of two estimators.

4.2 Previous Chemotaxis Techniques

How is biology efficient at chemical tracking and what principles can we learn from biology to help us with engineering design? Currently, mammalian olfaction is extremely complex and while many components are known, our understanding of the mechanism is scratching the surface. On the otherhand, there are many studies of single-cell mobilization in chemical gradients, called chemotaxis. We examine two chemotaxis and navigational techniques inspired by these mechanisms.

At the most fundamental level, chemical tracking is essential to primitive organisms. Humans have five senses, some highly evolved, whereas single-celled organisms essentially have two, touch and smell/taste. Without either, the organism would not be able to hunt its food and eat, or avoid predators. Thus, a single-cell must perform the computation necessary to achieve survival. It integrates its senses in chemotaxis, the process of mobilizing in a chemical gradient. We will now examine previous chemotaxis random walk locomotion and receptor clustering, and the associated algorithms inspired by these

mechanisms:

1. A single-sensor biased random walk and a two-sensor directional sensing algorithm for gradient tracking.

2. Multiple biased random walks for tracking multiple sources.

First, bacterial chemotaxis principles are reviewed. Then, the two random-walk algorithms are discussed, and they show how this strategy can be used in single-node and multi-node cases.

4.2.1 Bacterial Chemotaxis Principles

Figure 4.1. Example of a chemotaxis run and tumble trajectory, or random walk behavior. 30 s in the life of one Escherichia Coli K-12 bacterium swimming in an isotropic homogenous medium. The track spans about 0.1 mm, left to right. The plot shows 26 runs and tumbles, the longest run (nearly vertical) lasting 3.6 s. The mean speed is about 21 mm/s. A stereoscopic view can be seen in Bergs paper [17].

Chemotaxis is the mechanism by which an organism mobilizes in a chemical gradient. A single-cell is known to integrate information from its receptors, or chemical sensors, to control its movement through the flagella. The behavior of bacterial chemotaxis can be characterized in two ways: 1) a run and 2) a tumble phase. This is dictated by the rotation of the flagella, the motor movement, on the organism. When the counter-clockwise rotation aligns the flagella into a single rotating bundle, the bacterium swims in a straight

line, known as the run phase. When the clockwise rotation breaks the flagella bundle apart such that each flagellum points in a different direction, the bacterium to tumbles in place, known as the tumble or rotational phase. The bacterium alternates these two phases to move, using relatively straight runs interrupted by random tumbles that reorient the bacterium (illustrated by Fig. 4.1). With no gradient present, this movement exhibits a random walk behavior (see Appendix C for a mathematical description of a random walk). With a gradient present, the cell will start to exhibit longer runs in the direction of the gradient before tumbling and will tumble sooner if it finds it is going in an orthogonal direction to the gradient. This behavior exhibits a biased random walk, utilized in Kadar and Virks [49] and Dhariwals [24] algorithms. It is thought that this biased random walk provides directionality to the organism while keeping the organism flexible enough to find other sources (i.e.: prevents the organism from getting caught in a local minimum) [68].

Signaling in E. Coli chemotaxis relies upon protein phosphorylation. The key enzyme in the pathway is a histidine kinase (CheA), the activity of which is modulated by binding of chemoeffector to receptors and by the level of receptor methylation [95]. Changes in receptor methylation levels result in sensory adaptation, enabling the cell to detect further changes in concentration as it swims in chemical gradients. This is similar to when our visual system adjusts to low-light levels so that we can detect subtle differences. Receptor methylation also acts as a short-term memory of the chemical environment, utilized in Dhariwal et al.'s algorithm [24] and a by-product of Rosen/Hasler's [82]. In addition, it was observed a little over ten years ago that chemotaxis receptors form clusters at cell poles in E. coli. Since then, chemoreceptor clustering has been demonstrated in all bacteria and archaea that have been examined to date [33]. Moreover, it has recently been shown that all other chemotaxis proteins in E. coli localize to the cluster of receptors [60] [96] [10], thereby forming a large sensory complex. It is hypothesized that this receptor clustering helps to increase specificity and convergence time in localizing the chemical gradient. This type of mechanism is used in Rosen and Haslers work [82].

4.2.2 Single-Node Biased Random Walk and Receptor Cooperation

Since chemotaxis is an efficient navigation strategy in gradients, engineers have designed algorithms based on this mechanism to localize diffusive sources. An initial approach is by Kadar and Virk [49]. They compare a directional sensing algorithm they call chemotaxis to a biased random walk model. In the terminology presented in the previous section, the biased random walk is chemotaxis movement while their chemotaxis algorithm is a type of receptor cooperation. To keep terminology consistent, these algorithms are notated as *biased random walk* and *receptor cooperation* respectively. The authors use a $f = (1/r)^2$ decay for the gradient field for the region $0 < r < 5$. The additive noise is uniform random variable from [-0.5,0.5]. All the examples are conducted on a fixed grid composed of *units*. The *organism* is placed (4,3) units away from the source. In the noise regimes, the initial signal-to-noise ratio (SNR) can be computed from:

$$SNR = 10log_{10} \mid \frac{A_{signal}}{A_{noise}} \mid \tag{4.1}$$

where the organism is 5 units from the source so the $A_{signal} = f = (1/5)^2$ and A_{noise} $(noisestandarddeviation) = 1/sqrt(12)$ (standard deviation for a uniform random variable). Plugging this into (4.1), there is a starting SNR of -17.17 dB. The *biased random walk algorithm* makes all decisions from current time measurements and no short-term memory is assumed. For each step,

1. A run phase is executed. The run speed increases as it becomes closer to the source but slows as it hones in on the source:
   (a) $>$ 10 units from source, the step size is 0.5 units, thus the optimum steps to the source is 8.
   (b) Between 1 and 10 units, the step size is $0.5 + f/2$.
   (c) $<$ 1 unit, the step size is $1/f$
2. The tumble phase rotates the organism. The angle direction is the previous angle plus a uniformly chosen random variable from -28 to 28 degrees.

Table 4.1. Comparison of Kadar and Virks algorithms, averaged over five Monte Carlo runs.

	Receptor Cooperation	Biased Random Walk
Stable Field	13.4	130
Noisy Field	>1000	129.2

The *receptor cooperation algorithm* uses a fixed step size but uses spatial information to gain information about the direction of the source:

1. The step size is a fixed 0.5 units, thus the optimum steps to the source is 8.

2. The positive direction of the source is computed from the two receptors on either ends of the cell (0.4 units).

3. The angle direction to progress in is chosen uniformly random from three choices of 0 or ± 14 degrees towards positive direction of the source (angle which the two sensors create a line).

The results of the algorithm are summarized in Table 4.1. In a stable gradient field, 1) the receptor cooperation algorithm localizes the source directly and quickly and 2) the random walk algorithm is indirect and slow. In the noisy field, 1) the receptor cooperation algorithm diverges and is not likely to reach the source and 2) the random walk algorithm performs similarly to the stable case. So while the receptor cooperation algorithm breaks down quickly in the presence of noise, the biased random walk algorithm is the same despite the noise level.

4.2.3 Multi-Node Biased Random Walks for Source Tracking

Dhariwal et al. further investigates the biased random walk aspect of chemotaxis but for multiple tracking nodes and sources [24]. Rather than assuming that an organism varies its run and tumble ratio depending on the concentration level as in Kadar and Virk, Dhariwal et al. assumes that the chemotaxis mechanism is based on short term memory that is able to detect a positive or negative gradient by comparing the current concentration to the last

locations concentration. This short-term memory has been verified in biology literature [31]. On a 2000×2000 unit grid, 100 robots are placed randomly using a uniform random distribution. In biology, this can be paralleled to a colony of bacteria. The speed of each robot is assumed to progress at a unit/second and each time step is a second. The robot mean free path(MFP), or run-length without bias, is 10 units. The source(s) are always assumed to be a circular disc with a radius of 5 units, but two types of gradient source models are used. The first model uses m sources placed randomly on the grid and modeled by an inverse square law:

$$Intensity(x, y) = \frac{1}{K} \sum_{i=0}^{m} \frac{q_i}{r_i^2} \tag{4.2}$$

The intensity can be sensed at a point (x, y) on the grid in the presence of m gradient sources, q_i is the intensity of the source S_i, K is a constant of proportionality and ri is the distance between the grid point (x, y) and the center of source S_i. The second source model assumes that the source decays over time, such as an impulse source with infinite boundaries or actual consumption of the source where the chemical is a nutrient that can be eaten by the bacterium nodes. The intensity of the source, S_i, at any time t is given by:

$$q_i(t) = (q_i(0)e^{-k_1 t} - k_2 \sum_{j=0}^{t} N_{ij}$$

where $q_i(0)$ is the initial intensity of S_i, k_1 and k_2 are constants which depend on the type of source, N_{ij} is the number of robots at source S_i depleting its energy at time j. This is used in conjunction with (4.2) to create an intensity map based on decaying. The run-and-tumble strategy used by each robot has three phases: move bias-length in previous direction, tumble, run, and repeat. It can be described with the following pseudo-code (each run time-limit is 5×10^4 seconds and 10^4 Monte Carlo runs were averaged to get the final convergence results):

WHILE NOT at Gradient Source OR Time-limit

IF ((Concentration_new AND Concentration_old exist) AND Concentration_new¿Concentration_old))

```
        biaslength= bias*MFP;
        MOVE biaslength in previous direction;
    END
    Concentration_old=I(x,y);
    tumble=random direction from choice of eight neighboring gridponts;
    FOR 1 to runlength
        MOVE to next point on grid in the tumble direction;
        timestep=timestep+1;
    END
    Concentration_new=I(x,y);
END
```

In Kadar and Virk, the bias is based on the concentration level at the current time. In Dhariwal, there is a bias if the concentration is positive (determined from a short-term memory), and the actual concentration intensity does not affect the bias.

In Figure 4.2.3, a scenario is run for the 100 nodes placed 900 units away from a single source. With no bias, there is little progress after 50,000 seconds, but with just 10% bias, every node is able to localize the source within, 40,000 seconds, and 80% of the nodes reach the source within 25,000 seconds. With a 40% bias, 80% are able to reach the node in 5000 seconds.

The 100 nodes are also tested for finding multiple sources. It is unknown how distant these sources are from each other, but they are introduced at different times with the same amplitude, and it takes about 5000 seconds for 10% of the nodes to reach each one after it activates and quickly decays.

Also, an error is placed on the gradient decision function to see how performance would degrade. In all biases, the gradient measurement is subject to a percentage of error (e.g. for the 6% error case, if the gradient is positive, there is a 6% chance it will be measured as

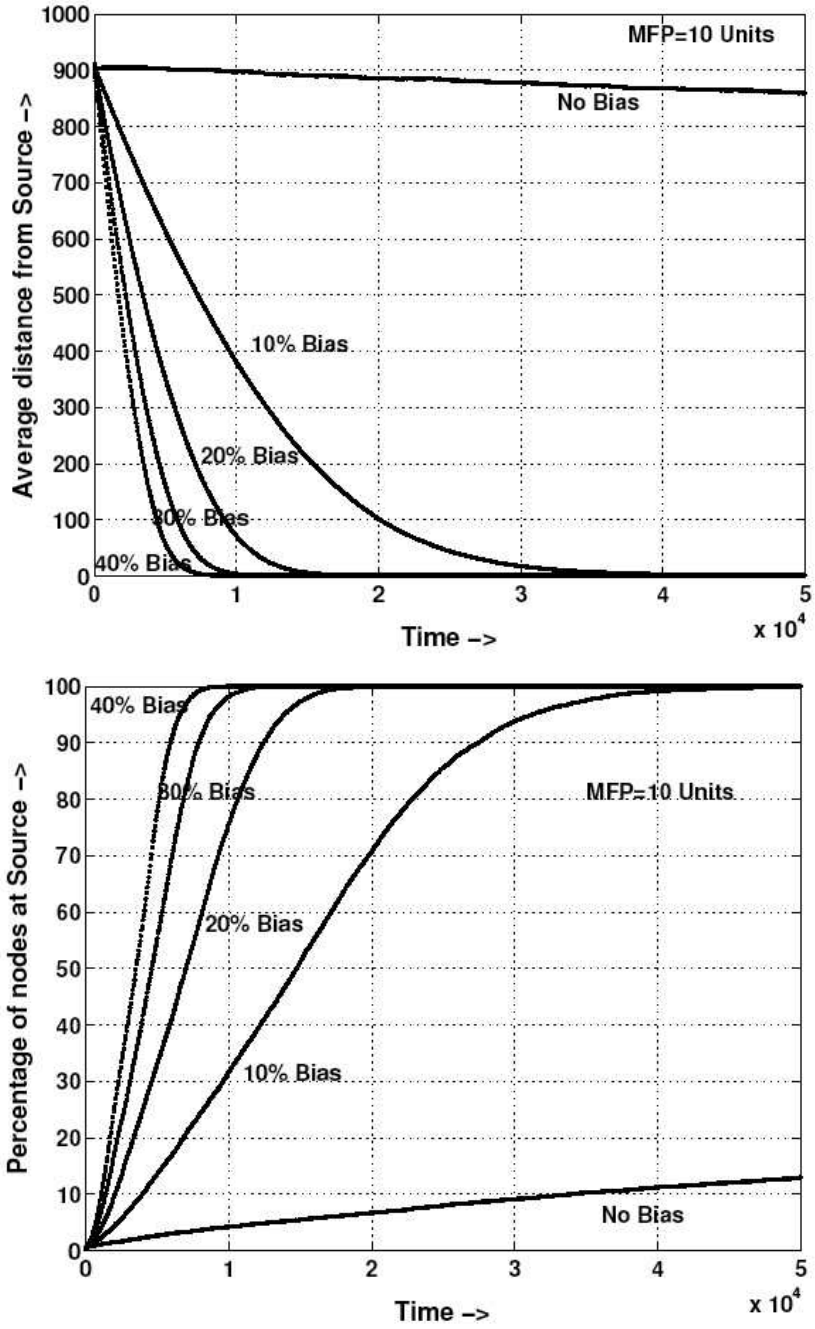


Figure 4.2. Increasing the bias decreases the time to convergence for this algorithm shown in a) the average distance between the robots and the source vs. time, and b) the percentage of robots at the source vs. time. Note there is just an inverse relationship between the two [24].

negative). In this scenario, the nodes still converge to the source but at a slower rate. The 20% error case takes about 50,000 seconds for all nodes to localize the source as opposed to the no-error case of 40,000 seconds. So, for full convergence, it takes about 20% more

time to converge. A similar trend is seen with the 40% error case, and it is expected to take around 40% longer to converge fully.

The single-source case is also expanded to a disc of 45 units, and the algorithm is shown to perform well for boundary detection.

4.3 Overview of other Chemical Source Localization approaches

The approaches that are array signal processing and robotics strategies use complex computation for multi-sensor localization. Strategies based upon bacterial chemotaxis are simple, but they have only been implemented for the single-sensor case. Our approach [82] is to utilize biological learning and the way cells mobilize in a chemical environment (Fig. 1.1) to improve a sensor array's localization and tracking of a diffusive source (heat or chemicals) while maintaining low-complexity for implementation [84]. In this work, we show how a Hebbian learning algorithm can be modified to approximate chemotactic behavior and compute the weights of an auto-associative network that can be used to determine the source location.

For the rest of this chapter, we present models of the physical environment, sensor array, and sensor measurements (Section 4.4), the classical Hebbian learning algorithm and connected auto-associative networks (Section 4.5), our modified Hebbian learning algorithm with controlled sensor cooperation and direction-of-arrival determination (Section 4.6), simulation results for a mobile array in various SNRs (Section 4.7), design and issues of implementing an array in hardware (Section 4.9), and the results of the stationary implementation in various environments (Section 4.10).

4.4 Model: Gradient Field and Sensor Array

A chemical field is dynamic in nature if turbulence (caused by wind, etc.) and noise (such as molecular Brownian motion) are taken into consideration. Excluding all these factors, we only model molecular diffusion, and obtain the result that the concentration, C (moles/cm^3),

from a continuous release point source in three-dimensions is the solution to Fick's 2nd Law [75]:

$$C(r,t) = \frac{\mu}{4\pi Dr}\text{erfc}(\frac{r}{2\sqrt{Dt}}) \tag{4.3}$$

where μ is the chemical release rate (moles/s), and D is the diffusion constant (cm^2/s). The parameter r is the radius from the point source, and t is the time from the initial release. In our model, the diffusion field of interest is the field at long diffusion times ($t \to \infty$), so the dimensionality of (4.3) is reduced to:

$$C(r) = \frac{\mu}{4\pi Dr}$$

Because the release rate varies greatly in nature (and for ease of use), we set $\frac{\mu}{4\pi D} = 1$. Therefore, the ideal source is modeled as $C(r) = 1/r$ for the diffusion field. Although this is a 3-D diffusion field, we only treat cases where the sensor array traverses a planar slice of this field.

Our sensor array is modeled as follows. Each sensor, $v_k[n]$, is the k^{th} input of N inputs at time sample n which measures the concentration signal, $C_k = 1/r_k$, at a distance r_k away from the source. The sensors are assumed to take measurements which are contaminated with independent and identically distributed (i.i.d.) noise:

$$\mathbf{v}[n] = \mathbf{c}[n] + \mathbf{n}[n] \tag{4.4}$$

where $\mathbf{c} = [C_1, C_2, ..., C_N]^T$, $\mathbf{n} = [n_1, n_2, ..., n_N]^T$, $\mathbf{0}$ is an $N \times 1$ vector of zeros, $\mathbf{n} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$, and $\mathbf{\Sigma} = \sigma^2\mathbf{I}$ where $\mathbf{I}$ is an $N \times N$ identity matrix.

We examine systems in biology to design a two-dimensional array. The mechanism to sense an odor or chemical gradient exists from the microbiological level all the way to mammals. Single-celled organisms which have an irregular oval-like shape provide inspiration for the sensor array shape. These animals' chemoattractant (chemical-sensing) receptors are uniformly distributed on their membrane in equilibrium. By this analogy, a circular array should be a good approximation to these organisms. For a circuit board

approximation, we choose a square array with an equidistant distribution of sensors on the perimeter.

4.5 Classical Hebbian Learning

Localization algorithms have been designed using off-line, complex sensor array processing algorithms, but we want to implement an algorithm for real-time operation and in the future, for low-power electronics. Thus we want relatively low complexity for large performance improvement, and biology can offer insight into such designs because many biological systems perform powerful computation while operating at low power.

Thus, we turn to neural computation (e.g. receptor signalling performed in chemotaxis) for inspiration. Hebb's rule [76] is a classical learning technique that adapts a set of weights to the input signal. For our purposes, we want to learn the connection strengths (correlations) between sensors to determine what direction a source is coming from. A sensor with higher correlation to other sensors is one that gets a higher amplitude on input and is, therefore, closer to the source.

A discrete-time matrix form of the Hebbian learning rule can be expressed as:

$$\begin{aligned}\mathbf{W}[n+1] &= \mathbf{W}[n] + \eta \mathbf{R}_{xx}[n]\mathbf{W}[n] \\ &= (\mathbf{I} + \eta \mathbf{R}_{xx}[n])\mathbf{W}[n]\end{aligned}$$

where $\mathbf{x}[n]$ is a vector of N inputs at time n, $\mathbf{W}$ is a $N \times N$ matrix of adaptive weights, $\mathbf{R}_{xx}[n] = \mathbf{x}[n]\mathbf{x}^T[n]$ is the correlation of the inputs, $\mathbf{x}[n]$, and η is a constant [76].

The change in $\mathbf{W}$ over a time period is proportional to the average of the input correlation matrix, $\triangle\mathbf{W} \sim \frac{1}{N}\sum_{n=0}^{N}\mathbf{R}_{xx}[n]$. Therefore, each element, w_{ij}, can be viewed as how well the ith sensor input correlates with the j^th sensor input. The η introduces a learning rate and short-term memory to the system. As a result, $\mathbf{W}$ can be viewed as the neural connections between each sensor and will retain memory of their connections for a short period of time. A graphical illustration of an auto-associative Hebbian network is given in Fig. 4.3. The mutual connection between sensors is analogous to sensor cooperation found

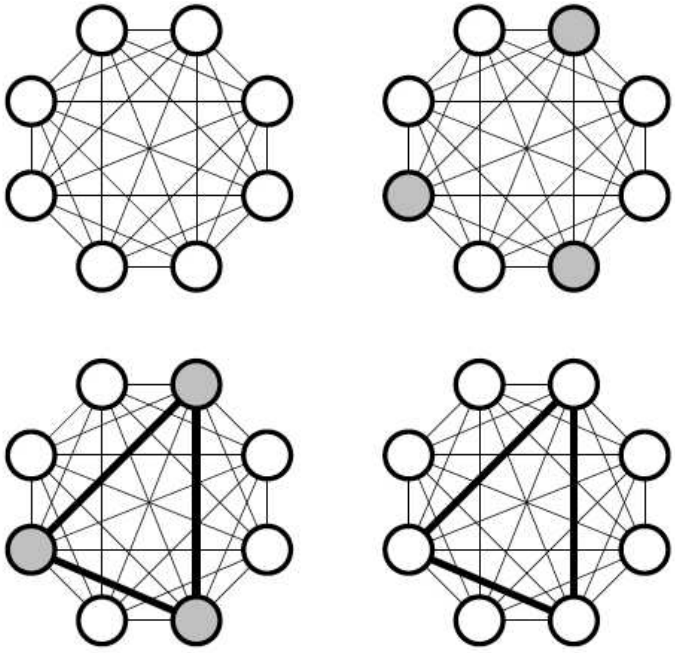


Figure 4.3. A simple Hebbian fully connected auto-associative network. When three of the units are activated by an outside stimulus their mutual connections are strengthened. The next time some of them are activated they will activate each other.

in biology. The difference is, in the Hebbian matrix adaptation, the sensors are fixed and the connections between them are adapting while in biology, the receptor locations adapt.

4.6 Modified Hebbian Learning for Localization and Tracking

In this section, a more descriptive version of Hebbian learning is presented, and we describe how our constraint affects the algorithm and the determination of the direction-of-arrival. We call our inputs, **v**, and they are correlated to a weighting/steering matrix, **A**. For each time step iteration, *n*, the output of the array, **y**, is computed as:

$$\mathbf{y}[n] = \mathbf{A}[n-1]\mathbf{v}[n] \tag{4.5}$$

where $\mathbf{A}[0] = \mathbf{A}_{\text{init}}$ (4.7). The matrix $\mathbf{A}_{\text{init}}$ has a dual role as the initial **A** as well as constraining **A** on each iteration (4.6).

The Hebbian learning algorithm is then used to update the steering matrix:

$$\mathbf{A}[n] = \mathbf{A}[n-1] + \eta\, \mathbf{v}[n]\mathbf{y}^T[n]$$

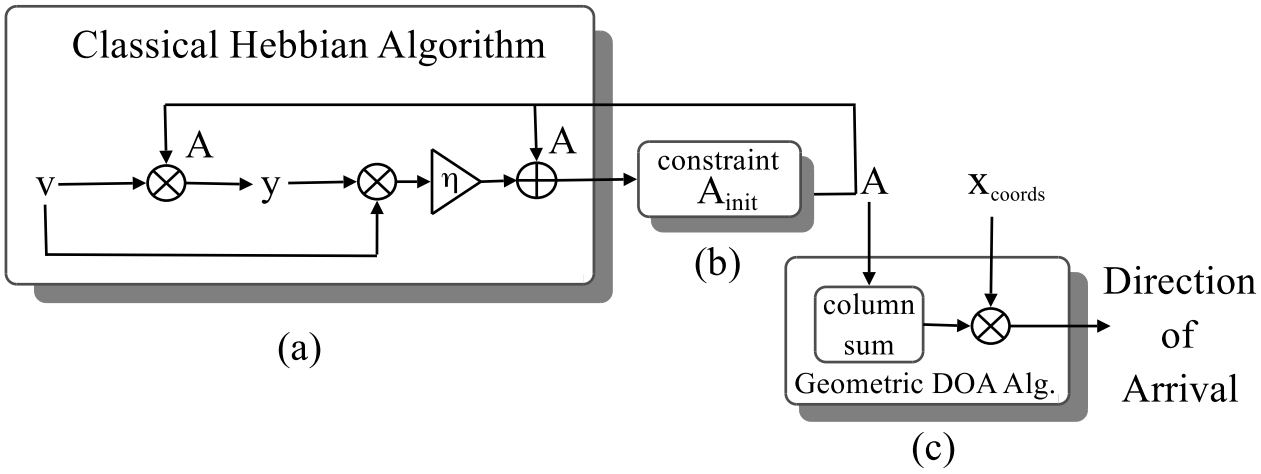


Figure 4.4. Diagram of Hebbian Learning Algorithm modified for control of sensor cooperation. The vector v contains the sensor inputs, the matrix A are the adaptive weights, η is the adaptation constant, and $\mathbf{x}_{coords}$ are the $[\mathbf{x}_{\text{coords}}, \mathbf{y}_{\text{coords}}]^T$ coordinates of the sensor array. a) Classical Hebbian learning updates the A matrix. b) Each element of A is multiplied by each element of the constraint, $\mathbf{A}_{\text{init}}$ to restrict the amount and strengths of the sensor connectivity. c) Each sensor's connections are summed into a total weight which then weights the sensor coordinates to determine the direction of arrival (DOA).

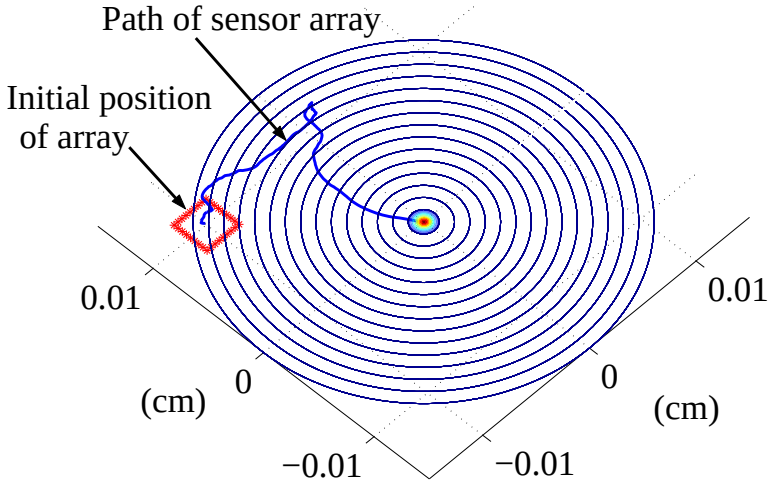


Figure 4.5. Example navigation path of a 32-element sensor array, $S_c = 5$, -1 dB starting signal-to-noise ratio (SNR). Source location occurred in 208 steps.

where $\eta = ((\mathbf{v}^T[n]\mathbf{v}[n])^{-1}$. A concise view of the Hebbian algorithm with added constraint and source angle determination is shown in Fig. 4.4.

On each iteration, a constraint that controls the sensor interconnectivity is imposed on **A**:

$$\mathbf{A}[n] = \mathbf{A}_{\text{init}} \circ \mathbf{A}[n] \tag{4.6}$$

where $\circ$ is an element-by-element multiplication and $\mathbf{A}_{\text{init}}$ is a circularly banded matrix with the band number corresponding to the sensor cooperation level, S_c:

$$\mathbf{A}_{\text{init}} = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & 0\ldots & a_{1N} \\ a_{21} & a_{22} & a_{23} & 0 & 0\ldots & 0 \\ 0 & a_{32} & a_{33} & a_{34} & 0\ldots & 0 \\ 0 & 0 & a_{43} & a_{44} & a_{45}\ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \\ a_{N1} & 0 & \ldots & 0 & a_{NN-1} & a_{NN} \end{pmatrix}. \tag{4.7}$$

In this example, $S_c = 3$ meaning each sensor and its nearest neighbors form the output (4.5) for a direction. The connections seen in Fig. 4.3 would be limited to the nearest $S_c/2 - 1$ neighbors.

This is directly related to how chemoreceptors cooperate for chemotaxis, the mechanism by which a cell senses and responds directionally to a chemical gradient (see Fig. 1.1). When a chemical binds to the receptors on the membrane of the cell, several receptors in a region signal a neuron. If all these receptors have chemical binds, the neuron, or weight, receives a high neural spike. Each column vector in the **A** matrix can be viewed as the neural beam pattern. It has been shown that organisms use spatial sensing mechanisms to compare receptor stimulation among different parts of the organism and then move accordingly [26]. Also, it has been observed that a cell's receptors begin to cluster towards the gradient direction when the gradient is suddenly reversed [19]. We conjecture that this is due to the fact that the organism wants to increase selectivity, or its beam pattern, in that direction. We parallel this spatial clustering behavior to what is known in the array signal processing literature as beamforming [48]. So, instead of moving the sensors to increase directional selectivity, we adapt the steering matrix.

The $\mathbf{A}_{\text{init}}$ is the key modification of the Hebbian learning algorithm which limits the amount and strength of the connections, or weights, in **A**. Sensors closer to the source have great impact on the computation of the source direction while those farther away have

less influence. Since all the connection weights to/from a sensor are summed to get the directional estimate, our constraint allows us to limit/attenuate side sensors for a particular column, which helps us control the learning algorithm's directionality and focus.

Effectively, we use three forms of $\mathbf{A}_{\text{init}}$: "Form 1" is the case of no sensor cooperation:

$$\mathbf{A}_{\text{init}} = \begin{pmatrix} 1 & 0 & 0 & 0 \ldots & 0 \\ 0 & 1 & 0 & 0 \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \\ 0 & 0 & 0 & 0 \ldots & 1 \end{pmatrix}$$

"Form 2" is the case sensor cooperation with no side connection attenuation (example of $S_c = 3$):

$$\mathbf{A}_{\text{init}} = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

For "Form 3", the weights of $\mathbf{A}_{\text{init}}$ have the following structure (example of S_c=3):

$$\mathbf{A}_{\text{init}} = \begin{pmatrix} 1 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\ \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} & 1 \end{pmatrix}$$

This form now places less emphasis on sensors contributions further away from the particular focussed direction (4.5). Each band S_c away from the diagonal has a $1/2^{S_c}$ weight.

Also, to keep the weights sensitive to input changes, $\mathbf{A}$ is bounded by $\eta\|\mathbf{v}\| \leqq \|\mathbf{A}\| \leqq \|\mathbf{v}\|$.

The direction of the sensor array movement is calculated using the steering matrix. The center of the square sensor array is (r, θ) away from the source. This center coordinate vector can be averaged from all the individual sensors' distances, r_k and angles, θ_k from the source: $[r, \theta]^T = \frac{1}{N} \sum_{k=1}^{N} [r_k, \theta_k]^T$. We now represent the center with cartesian coordinates $\mathbf{x} = [x, y]^T$, and the sensor coordinates in reference to the center are:

$$\mathbf{X} = \begin{bmatrix} x_1 - x & x_2 - x & \dots & x_N - x \\ y_1 - y & y_2 - y & \dots & y_N - y \end{bmatrix}^T = \begin{bmatrix} \mathbf{x}_{coords} \\ \mathbf{y}_{coords} \end{bmatrix}.$$

Next, the direction of the source from the centroid of the array is estimated as:

$$\mathbf{d}[n] = \mathbf{1}^T \mathbf{A}[n] \mathbf{X} = \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

where **1** is an Nx1 vector of ones. We can rewrite this equation as:

$$\begin{bmatrix} d_x \\ d_y \end{bmatrix} = \begin{bmatrix} \sum_i a_{ij} \cdot \mathbf{x}_{coords} \\ \sum_i a_{ij} \cdot \mathbf{y}_{coords} \end{bmatrix}$$

where $\cdot$ is an inner product. In other words, the columns of **A** are summed, where each element in a column corresponds to a weighting of a sensor's connection to itself and other sensors. We make the assumption that the sensor with the largest summed weighting will be the closest to the source. Most likely, this is true since it receives a higher input amplitude than the other sensors. Then each summed column weights each sensor coordinate and is used to create a geometric estimate of the source direction.

The new sensor array centroid coordinate is calculated as

$$\mathbf{x}[n+1] = \mathbf{x}[n] + \delta_{\text{fixed}} \cdot \mathbf{d}[n].$$

The iteration is stopped when the center of the array is within a fixed-step threshold of the source:

$$r \leqq \delta_{\text{fixed}} = \frac{1}{100} r_{\text{init}}$$

An example iteration of the algorithm is shown in Fig. 4.5.

4.7 Simulation Results from a Mobile Array

The primary goal of this experiment is to assess the spatial advantage gained by a mobile, square sensor array using three forms of steering constraint matrices (4.7) : 1) an identity matrix (only one chemoreceptor for one neuron), 2) a banded matrix with unity weights (multi-receptors each equally signalling a neuron), and 3) a banded matrix with $1/2^{S_c}$ bands (multi-receptors signalling a neuron where receptors further away from the neuron have less weight). Since organisms use visual cues or sensation to determine if they have reached a source, no mechanism was incorporated for the array to internally detect this. Detection is assumed when the array comes within δ_{fixed} of the source, and the steps/time for the detection to occur is called the localization time.

In the simulation, the center of the $3\mu\text{m} \times 3\mu\text{m}$ sensor array is placed $r_{\text{init}} = 141\mu\text{m}$ away from the source. Although the field is infinite at $(0, 0)$, the source detection threshold distance is set at a fixed step size, δ_{fixed}, which is $1/100$ of r_{init}. Therefore, the concentration level C_{init} at the initial array placement is $1/100$ smaller than the concentration at the source threshold, δ_{fixed}.

Each sensor array is characterized by the localization time to the target vs. starting signal-to-noise ratio (SNR). Starting SNR is defined as the initial average SNR of the sensor measurements (4.4): $\frac{1}{N}\sum_{k=1}^{N} 20\log_{10}(|C_k[0]/n_k[0]|)$. Also, the effect of sensor cooperation on the an array's localization time is assessed. The algorithm is tested over several parameters:

- N, the number of sensors, is varied by factors of 2 over 4, 8, 16, and 32.
- S_c, the sensor cooperation level, is run for odd numbers from 1 to $N - 1$.
- *startingSNR* is evaluated from approximately -8 dB to 8 dB in 2 dB steps.

The localization time is computed over all parameters; one thousand Monte Carlo iterations are computed for each combination.

In Fig. 4.6, a histogram of the amount of steps (or time if a velocity constant is given) for the array to reach the target is plotted. The distributions are top-heavy and have very

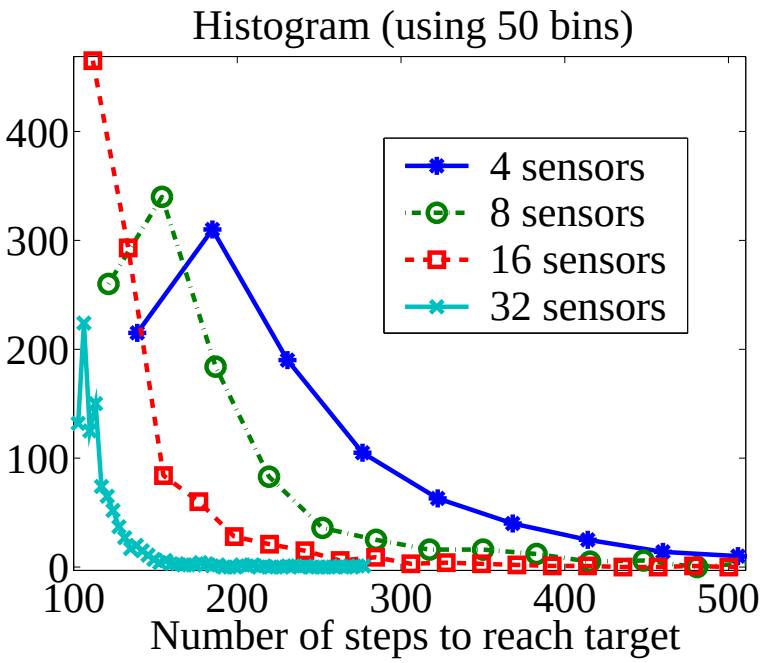


Figure 4.6. Distribution of localization time vs. sensor array size for 1000 Monte Carlo runs with approximately 4 dB of sensor starting SNR and no sensor cooperation. Some tails actually extend out to around 2500 steps but are truncated for illustration.

long tails. Due to limits on computation, if the number of steps exceeds 100K, the iteration is stopped and noted. In Fig. 4.7, the mean number of steps of the thousand iterations vs. SNR and the number of iterations stopped at 100K vs. SNR are shown for an 8-sensor array. From the plots, it is inferred that iterations with long localization times directly affect the mean statistic. One can see that when the number of iterations exceeding 100K in Fig. 4.7 b) grows, this behavior skews the mean computation in lower SNRs and causes it to deviate from its quadratic behavior on this graph. In Fig. 4.6, most iterations cluster around a short time value; therefore, the median is a better characterization of the heavy-tailed distributions and is our preferred statistic.

Assuming a measure-and-go strategy with δ_{fixed} as the distance the sensor array moves each step, the chemical source localizer converges to the source in 100 steps in the optimal case. As the SNR is lowered, the median number of steps to the source is used as a performance measure.

The performance gained with varying numbers of sensors and no sensor cooperation

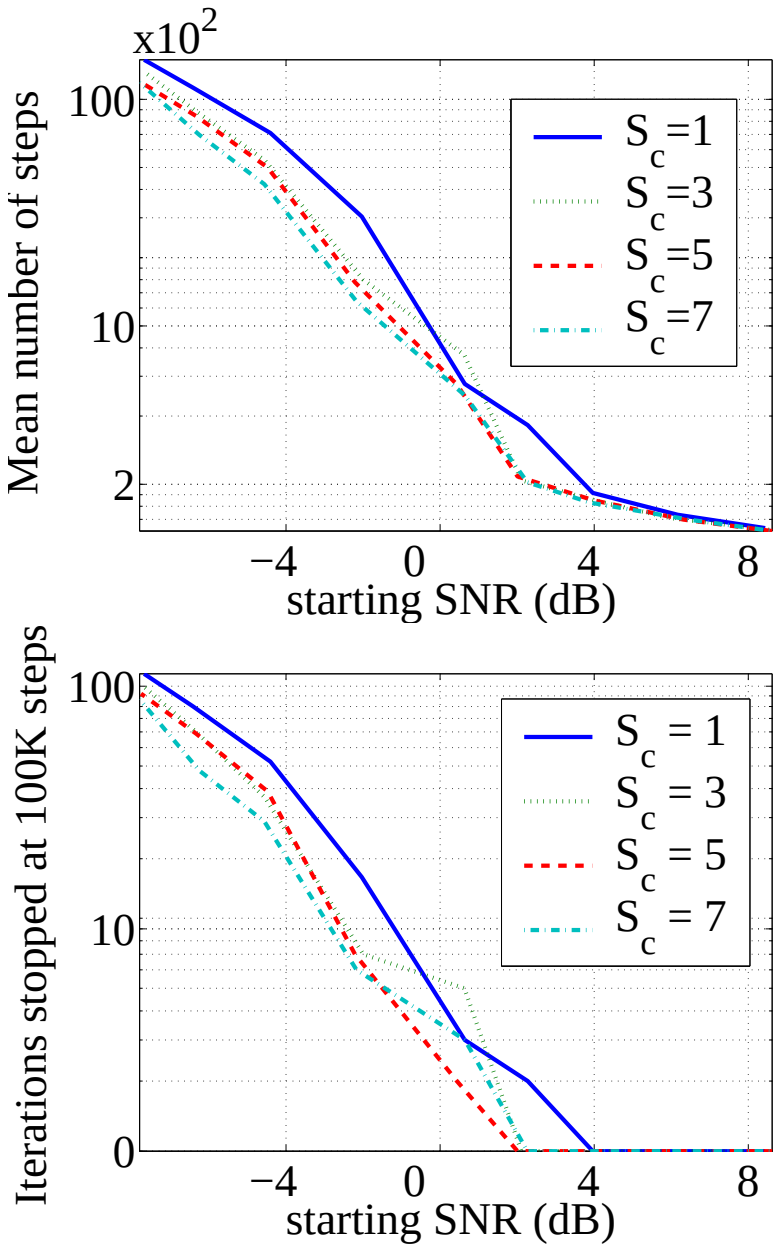


Figure 4.7. a) Illustration of the mean number of steps taken to localize the target vs. the starting SNR. b) Illustration of the number of iterations that were truncated at 100K steps vs. SNR. Outlying long localization times directly impact the mean of the steps. Therefore, the median is the preferred statistic.

($\mathbf{A}_{\text{init}}$ as an identity matrix (4.7)) is shown in Fig. 4.8. As expected, localization time increases as the SNR decreases, and it does so with a quadratic behavior. At a fixed SNR, the percentage improvement becomes less noticeable as the amount of sensors increases.

To simulate an equal-weight sensor cooperation case, the weights in $\mathbf{A}_{\text{init}}$ are set to 1 for the bands dictated by the sensor cooperation level, S_c. In Fig. 4.9, the higher the

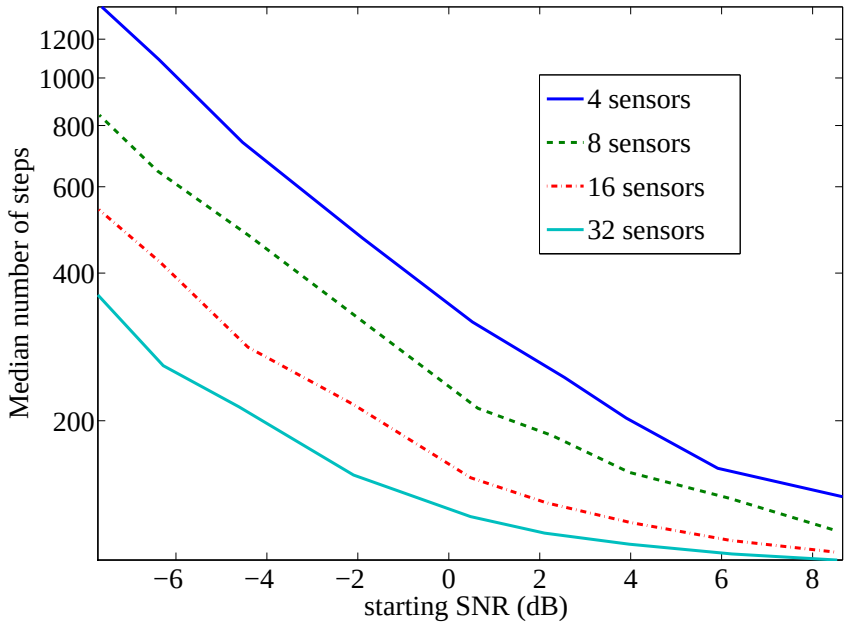


Figure 4.8. The effect of increasing the number of sensors on the localization time vs. SNR. 4, 8, 16, and 32 sensors are shown, and the SNR is varied from -8 dB to 8 dB. Due to the stepsize, the asymptotic lower bound is 100 steps.

sensor cooperation level for $S_c > 5$, the worse the performance is. In fact, the no sensor cooperation/identity matrix case, outperforms sensor cooperation for high SNR; in low SNR, $S_c = 5$ has the best overall performance. One can make sense of this from the directional pattern formed by the sensor cooperation. If a sensor on the upper right-hand corner observes a high concentration coming in from the upper right, then using its nearest neighbors' (2 on each side for $S_c = 5$) measurements in addition to its own will add extra information needed to gain better resolution of the angle in that direction. On the other hand, taking all sensors around the array and weighting their information equally for each direction will cause distortion and degrade the angle resolution as opposed to using clusters in each direction.

In Fig. 4.10, a comparison of the three $\mathbf{A}_{\text{init}}$ forms and their influence on performance are shown for 3 sensor array sizes. The unequally weighted $\mathbf{A}_{\text{init}}$ (where $S_c = N/2 + 1$ for each N) performs consistently better than no sensor cooperation for all SNR. The unity-banded $\mathbf{A}_{\text{init}}$ case (where $S_c = 3$ for $N = 8, 16$, and $S_c = 5$ for N=32) is worse than

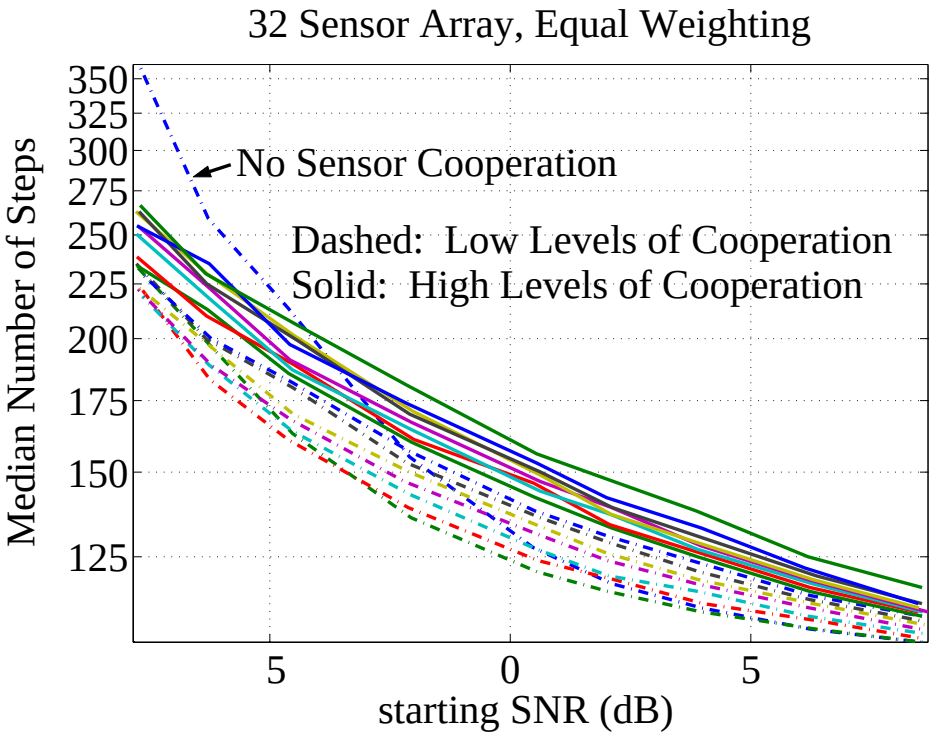


Figure 4.9. The $\mathbf{A}_{\text{init}}$ of Form 2 degrades performance as more sensor cooperation levels are added to a 32 sensor array. (The lower levels of sensor cooperation correspond to A with less than $S_c/2$ bands.) The lower levels of sensor cooperation perform better than the higher levels in all SNRs, but not as well as no sensor cooperation in high SNR. The localization time vs. starting SNR is shown for the no sensor cooperation case and odd sensor cooperation levels between 3 and 31. Due to the stepsize, the asymptotic lower bound is 100 steps.

no sensor cooperation for high SNR, but for low SNR, this method significantly reduces localization time. A 16-sensor array using this method is comparable to a 32-sensor array with no sensor cooperation in -8dB SNR. But, the two methods have trade-offs. If SNR varies, it may be more desirable to use $\mathbf{A}_{\text{init}}$ of Form 3 to consistently reduce localization time; otherwise, if the sensor array only operates in low SNR conditions, Form 2, may be more desirable.

In Table 4.2, localization time of various sensor arrangements are numerically compared for two SNRs. The sensor cooperation algorithm using the third form of $\mathbf{A}_{\text{init}}$ is run for various N and S_c and is compared to the 4-sensor, no sensor cooperation, case. Increasing the number of sensors significantly improves performance while the conservative $\mathbf{A}_{\text{init}}$ of Form 3 helps with a localization time reduction of about 5 to 15%.

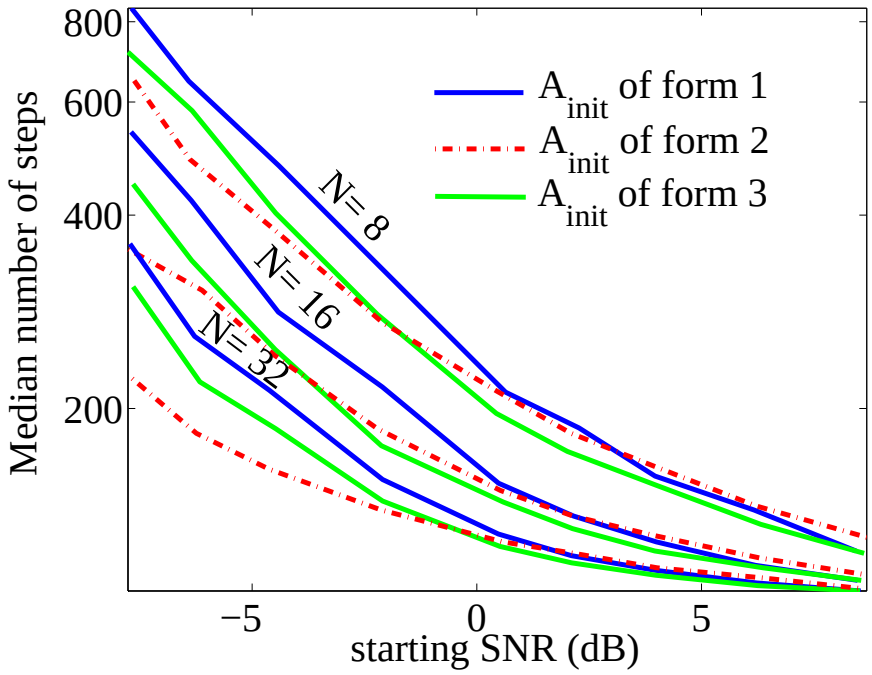


Figure 4.10. Comparison of effect localization time vs. starting SNR for the three forms of A_{init}. The forms are compared for 8, 16, and 32 sensor arrays. Form 3 performs better than Form 1 in all SNR while Form 2 performs much better than all algorithms in low SNR but performs slightly worse in high SNR. Due to the stepsize, the asymptotic lower bound is 100 steps.

Table 4.2. A comparison of the median steps (MS) for source localization in 0 dB and -7.5 dB for the single sensor mobile case, and a comparison of the $N/2+1$ banded A_{init} of Form 3 to the 4-sensor no sensor cooperation case.

N	S_c	MS for 0 dB (% improved)	MS for -7.5 dB(% improved)
1	1	11130 (-3071%)	35646 (-2511%)
4	1	351 (baseline)	1365 (baseline)
4	3	304 (13.4%)	1256 (8.0%)
8	5	210 (40.2%)	693(49.2%)
16	9	150 (57.3%)	435 (68.1%)
32	17	126 (64.1%)	301 (77.9%)

Just by using four sensors instead of one ($N = 1$), the localization task shortens by magnitudes as seen in Table 4.2. The single sensor case is based on a biological chemotaxis algorithm [49]. In the algorithm, the sensor moves randomly when the change in concentration gradient is negative else continues in the same direction if it is positive. The single sensor median number of steps is calculated from 50 Monte Carlo runs.

4.8 Performance comparison to previous chemotaxis-based techniques

We have reviewed: 1) a single-node biased random walk and receptor cooperation algorithm, 2) multi-node biased random walks, and 3) **our** multi-receptor clustering algorithm. The biased random walk is able to provide directionality while allowing enough randomness for the organism search out a global minimum. A good example of this case is in Dhariwal et al.s two-source scenario where various nodes are able to find multiple sources, and there is a shift of the percentage of nodes towards the larger source over time. The sensor cooperation algorithms are able to utilize the gradient information directly to navigate to a source. When local groups of sensors, or receptor clusterings, are fused to spatially smooth sensor information in addition to time averaging, an array of sensors is able to perform better in a noisy environment than when each sensor adapts independently. The receptor cooperation algorithm is useful for low SNR and low gradient scenarios to exploit the directionality out of sensor inputs. In Table 4.3, the parameters of each algorithm are categorized for comparison: the number of sensors, whether the sensors are independent or cooperative, the noise level, the optimum number of fixed steps to the source, and the number of steps to the source/localization time. With the multitudes of differences between each algorithm, it can be difficult to compare the performance between each algorithm. The step-size may be variable or not exist at all if the algorithm is continuous-time and not discrete-time. The chemotaxis-based algorithms use a fixed or minimum step size, so the optimum number of steps is reported, and the results are compared using a normalized time measure:

$$steps_{normalized} = \frac{steps_{actual}}{steps_{optimum}} \tag{4.8}$$

This may not be the best metric of such algorithms, since the localization time is the true metric. Since these algorithms can be implemented with any node velocity, it is difficult to compare these before implementation. A standardized set of performance metrics are much needed to compare the algorithms. The performance of these algorithms is compared

Table 4.3. Comparison of various parameters in each algorithm.

	# of sensors	Independent or Cooperative Nodes	Noise Level	Optimum steps for fixed step-size
Kadar/Virk	1,2	Independent/ Co-operative	-17.17 dB	8
Dhariwal et al.	100	Independent	0,6,20,40% error on binary gradient decision	9
Rosen/Hasler	4,8,16,32	Cooperative	-8 to 8 dB	100

Table 4.4. Performance comparison of the algorithms, showing the strategy, the number of sensors, the noise regime, and the localization time normalized by the optimum step size.

	Parameters	Normalized # of steps
Kadar/Virk	1 sensor biased random walk, -17.17 dB starting SNR	(130/8) 16.25
Dhariwal et al.	100 sensors, 20% error, (steps calculated from 50000 seconds divided by MFP=10)	(5000/90) 55.56
Rosen/Hasler	4 sensors, 3 sensor cooperation, -7.5 dB starting SNR	(1256/100) 12.56

in Table 4.4. The number of sensors used, the noise level, and the normalized number of steps is shown for each algorithm. An interesting note is that Kadar/Virks algorithm does very well for a low starting SNR. It is important to note that the results were only averaged over five Monte Carlo runs while 10^4 and 10^5 Monte Carlo runs were averaged in Dhariwal et al. and Rosen/Hasler respectively. So, the findings in Kadar/Virk may not have a sufficient statistics for this result. Nonetheless, their work was one of the first navigation techniques to try both the run-and-tumble and gradient following strategies. For the amount of nodes, Dhariwals algorithm takes much longer to converge to the source, but because of multiple nodes, the method has the advantage of finding multiple sources and even boundaries of chemicals, such as oil spills. In Rosen/Hasler, a 4-sensor array in low SNR obtained reasonable results for a single source, and this technique has the advantage of using a single node while enhancing performance via numerous sensors and algorithmic complexity. For example in Table 4.2, by quadrupling the number of sensors with sensor cooperation in this algorithm, the localization time can decrease by 3-fold.

4.9 Hardware Implementation for a Stationary Array

In this implementation, we focus on temperature localization due to high reliability of inexpensive temperature sensors though our techniques easily extend to chemical localization. Inexpensive temperature sensors are highly accurate and less prone to nonlinearities such as drift; chemical sensors have major problems with drift and can also saturate. Our bio array processing algorithm assumes linear ideal measurements with Gaussian error, therefore linear sensors are needed. Also, the algorithm assumes diffusion as dictated by Fick's 2nd Law, and both heat and chemical diffusion obey this. Thus, it is natural to try our algorithm first for heat localization, but this design can easily extend to tracking in chemical gradients as well.

Heat diffuses much more rapidly than chemical diffusion. In this section, we show that the heat from a +3000° C light bulb dissipates to 59° C in 10.4 cm, and at 22.6 cm away from the source, the temperature is 29° C, yielding a steep temperature gradient over space. At 30 cm, the heat from the lightbulb is negligible because the temperature is measured at approximately 25° C, which is room temperature. In this environment, we have a limited amount of space, and a mobile implementation would not be practical. A stationary implementation to locate the source at a near distance is more useful in a limited-space experiment. Therefore, our implementation is stationary. We simulated the stationary array, and the results are similar to the implementation results presented here. The difference is that in mobile simulation, Form 3 performs better than Form 2, but in the stationary case, Form 2 performs better than Form 3. The sensor cooperation methods (Forms 2 and 3) perform better than no sensor cooperation (Form 1) in both the stationary and mobile cases.

Our goal is to test implementation feasibility of 2-D gradient-source localization using one 4/8 multi-sensor array with our modified Hebbian learning for sensor arrays [82]. Two temperature sensor array prototypes were constructed: one yields noisy measurements (standard deviation (std) of $0.4°C$), while the other implementation has little noise (std of

$0.1°C$). This yields two platforms to test the algorithm on. Also, it is shown that as the sophistication of the algorithm increases, the better the circuit is able to localize and track real temperature measurements with substantial variation. Therefore, the algorithm could be used in a wide range of applications, and in particular, a sensor integration using a higher number of sensors to measure diffuse far-field sources.

4.9.1 Temperature Sensor Setup

Our circuit implementations uses the Hewlett-Packard Labs' Smartbadge, a microcontroller-based device used for sensor integration and computation [61]. Microchip TC74 temperature sensors quantize the temperature in the room to the nearest Celsius degree and interface to the Smartbadge. The circuit schematic is seen in Fig. 4.11. A photograph of the setup can be seen in Fig. 4.12. There is an $11c$m distance between each sensor for the 4 sensor configuration and a $5.5c$m distance for the 8 sensor configuration. The circuit's measurement time per sensor is 10ms. When implementing the above algorithm in a real system, sensor bias needs to be compensated, and the η parameter needs to be adjusted for the correct adaptation rate vs. sensitivity trade-off [84].

4.9.2 Heat Source Calculations

We calculate the ideal heat dissipation length from a lightbulb beginning with the thermal conductivity of air, $.025\frac{Watts}{meter*Kelvin}$. From [1], it is shown that a 60-W bulb radiates 73.3% of its power as heat, therefore the effective bulb wattage is $44W$. The temperature change over the gradient is $\Delta T = 2700K(2527°C) - 300K(27°C)$ where $2700K$ is the temperature of the lightbulb's tungsten filament and $300K$ is the measured room temperature. Using simple unit conversion, the bulb's heat will dissipate in air after 73cm, ideally.

For the first set of experiments, a 60-Watt incandescent light bulb lamp with a 12.7cm diameter is used as the heat source. The temperature 16cm from the lamp reached a steady-state of 35°C after 15 minutes through empirical studies. For Figs. 4.13 through 4.16, the source was placed at a near-field distance of 16cm from the center of the board at a $-90°$

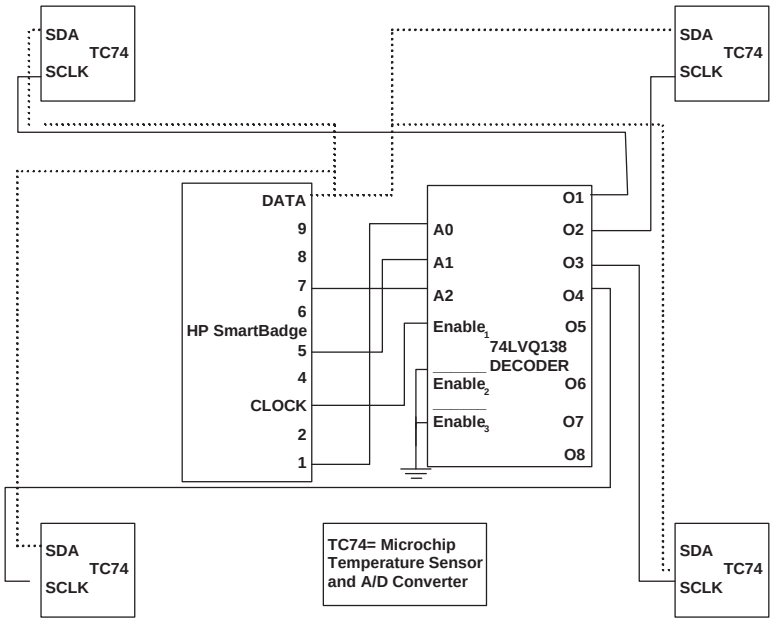


Figure 4.11. Schematic of sensor setup using the HP SmartBadge IV. Four sensor layout maximizes the perimeter of the board. A four sensor configuration is used in the first prototype, and both four and eight sensors are used in the second prototype.

angle.

For the second experiment configuration, a 100-Watt bulb was placed 15.75cm away from the center of the board. The steady-state values for each sensor scenario were recorded. In the diffusion-only scenario, with a heat source placed at a 100° angle between the first and second sensors, the 4 sensors reached 59°C, 53°C, 29°C, and 30°C in 5 minutes. With an oscillating fan between the third and fourth sensor under the same scenario, the sensors reached 36°C, 33°C, 25°C, and 25°C in 2 minutes with the first sensor oscillating between 38°C and 36°C in turbulence. For Figs. 4.18 through 4.20, the source was placed at a near-field distance of 15.75cm from the center of the board at a 100° angle.

4.10 Experimental results from stationary array

Our approach uses two computations as part of the tracking algorithm: a fixed sensor co-operation algorithm so each sensor effectively sees a particular direction, and an adaptive algorithm to track the particular source location. Fig. 4.13 shows the experimental system

Figure 4.12. Photograph of the implementation setup for the gradient source localizer. An incandescent lamp was used as the heat source. The sensors were interfaced to the HP Smartbadge through a controller, and a laptop interfaced to the HP Smartbadge was used to collect data.

tracking utilizing a memoryless weighting of the temperature measurements with the sensor coordinates. Fig. 4.14 shows the result of the algorithm with $S_c = 1$, which adapts each sensor output independently based upon the error signal. This approach results in significantly smoother tracking of the noisy temperature gradient signal. The approach further improves by increasing S_c to 3, which uses more spatial information for the algorithm (Fig. 4.15). Fig. 4.16 focuses the adaption by attenuating the side sensor's influences for each sensor direction when determining the geometric direction.

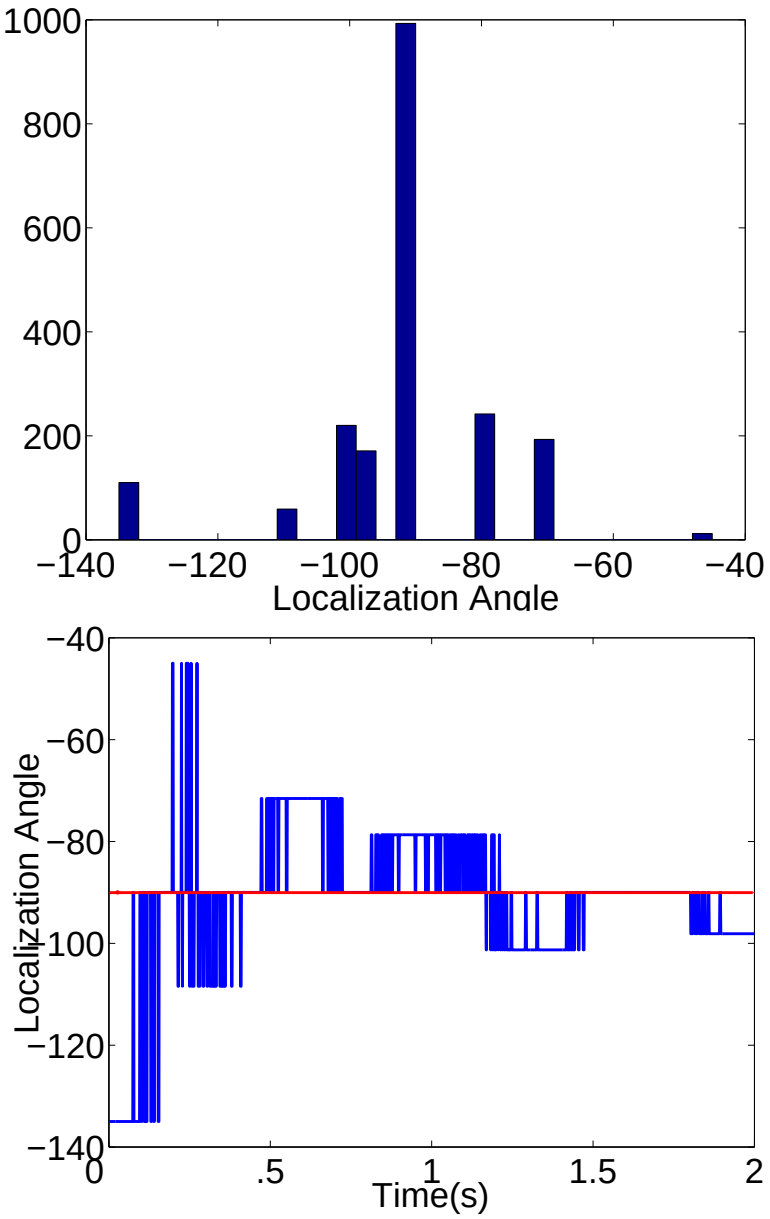


Figure 4.13. A simple memoryless (Form 1 without memory) algorithm based on 4-sensor temperature measurements. a) The mean angle is $-91°$**, median angle is** $-90°$**, and the standard deviation is** $14°$**. b) Within 150 iterations (0.15s), the source is localized. Even though there is high variance, convergence to the source angle is fast.**

4.11 Exploring Environmental Scenarios

Under the cleaner system for 4 and 8 sensors, three environmental scenarios are explored: a diffusive environment, a turbulent environment including wind gusts (setup seen in Fig. 4.17), and turbulence with added artificial Gaussian noise.

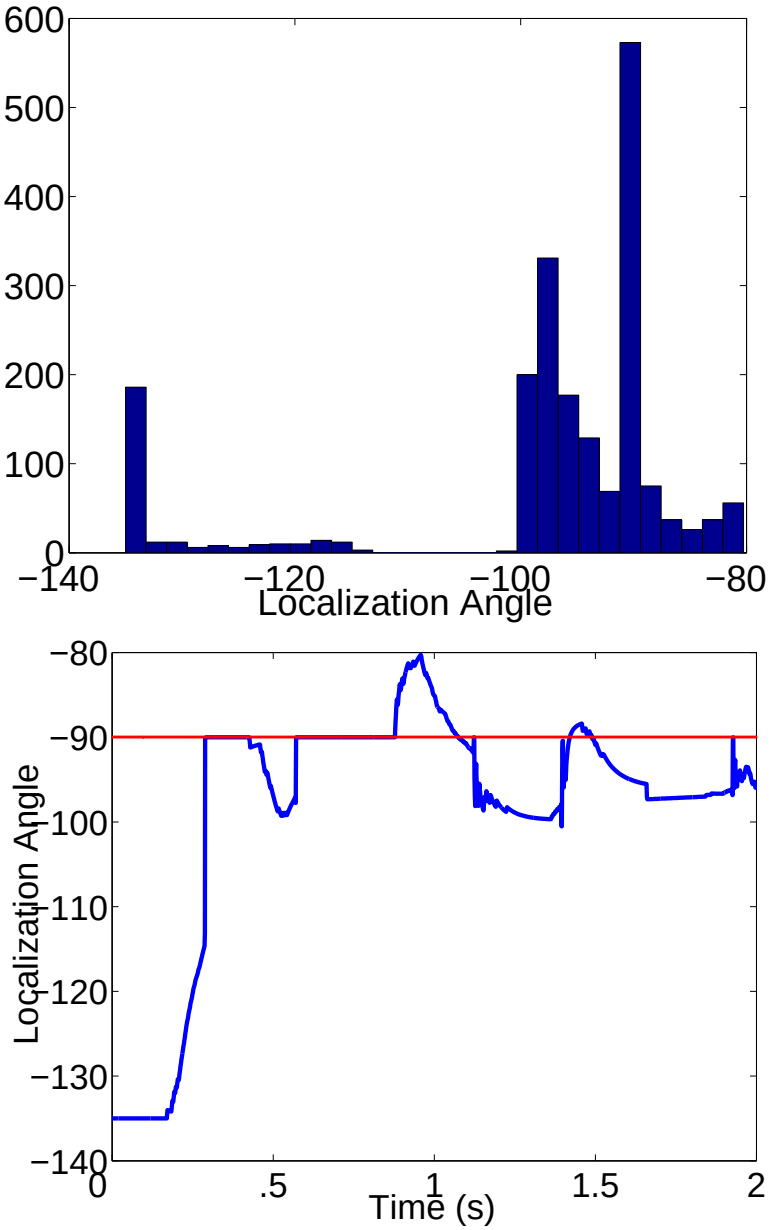


Figure 4.14. The 4-sensor algorithm with Form 1 A_{init} functions as a plain averager. a) The mean angle is $-98°$, median angle is $-95°$, and the standard deviation is $14°$. b) Within 300 iterations (0.3s), the source is localized. Note that accuracy begins to diverge with time.

4.11.1 Diffusive Environment Results

First, we run a 4-sensor system in a diffusive environment with the heat source placed at 100°.From the simple algorithm in Fig. 4.18, the clean system results in accurate measurements despite the jitter introduced by the sensor quantization error. All algorithms localize the source to within 5° in 0.5 seconds, but not all converge to the true mean at the same rate

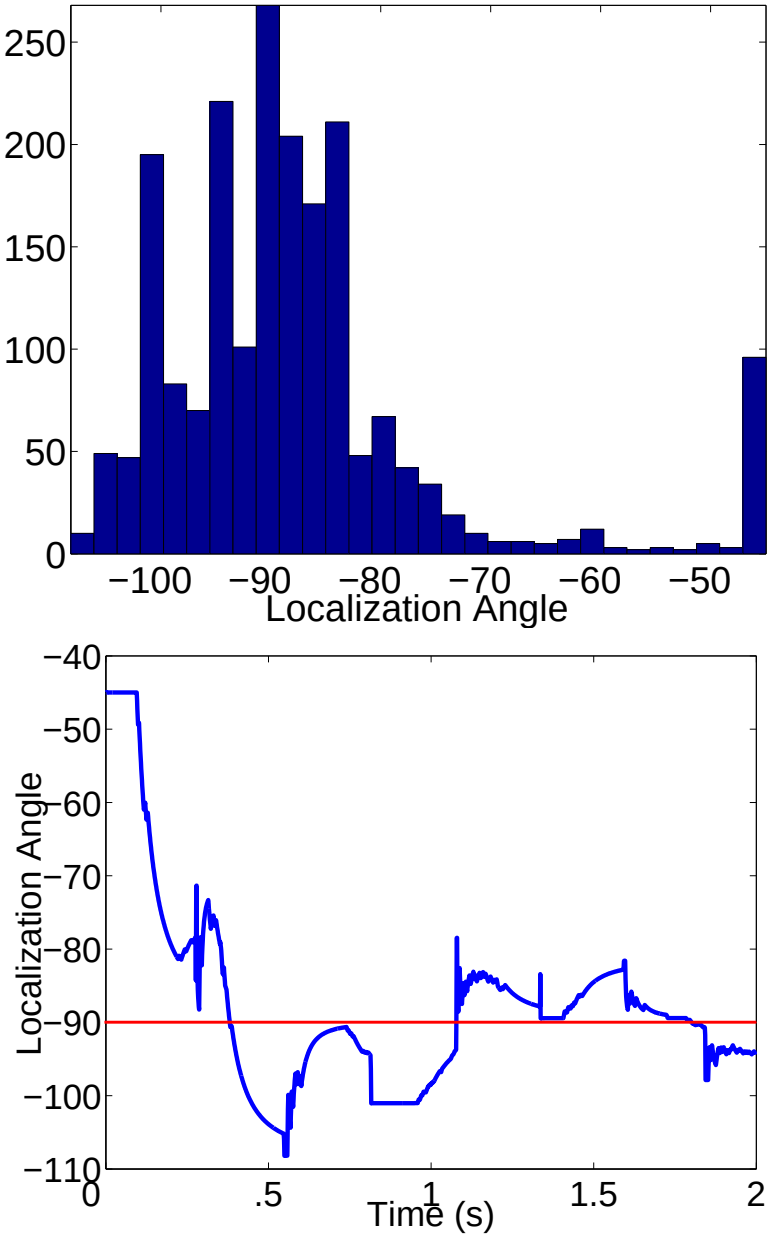


Figure 4.15. A_{init} **is of Form 2. a) The mean angle is** $-78°$**, median angle is** $-89°$**, and the standard deviation is** $13°$**. b) Within 700 iterations (0.7s), the source is localized. Note that accuracy improves with time, but convergence takes longer.**

and standard deviation. In Fig. 4.18 (a), the Form 1 $\mathbf{A}_{\text{init}}$, the averager, and the Form 3 $\mathbf{A}_{\text{init}}$, tapered tri-band, slowly converge to the angle, but the Form 2 $\mathbf{A}_{\text{init}}$, uniform tri-banded, perfectly tracks it. The sensor cooperation significantly improves the algorithm's response time to the heat diffusion. In Fig. 4.18 (b), the tracking curve is roughly quadratic, and the uniform 7-banded matrix responds the best. Again, the sensor cooperation shortens the

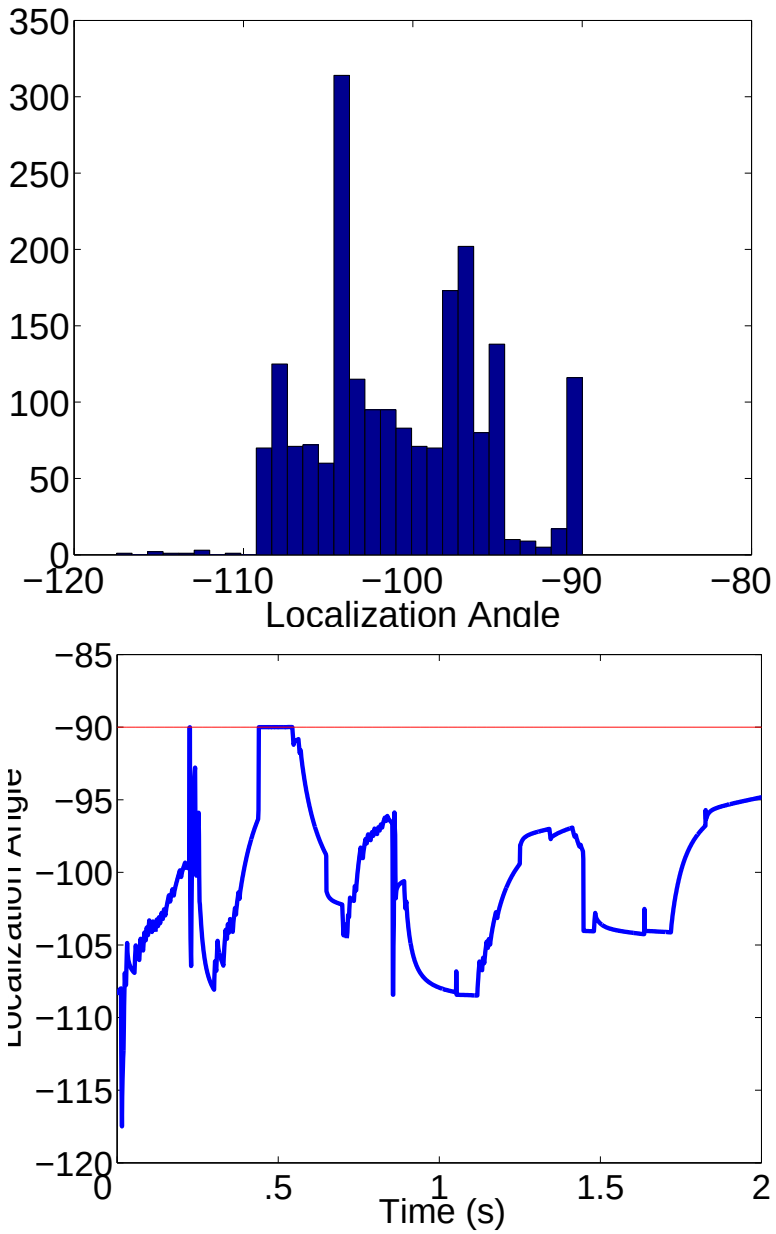


Figure 4.16. A_{init} **is of Form 3. a) The mean angle is** $-100°$**, median angle is** $-101°$**, and the standard deviation is** $5°$**. b) Within 500 iterations (0.5s), the source is localized. Note that its variance is significantly lower than the other methods, but the convergence takes longer. In two minutes, it does not fully converge, but the trend indicates that it will converge to** $-90°$**.**

convergence time.

4.11.2 Turbulent Environment Results

To simulate a turbulent environment, an oscillating fan was placed 20 cm away from the board at a $-90°$ angle. This caused wider variation in the source measurements, seen in the

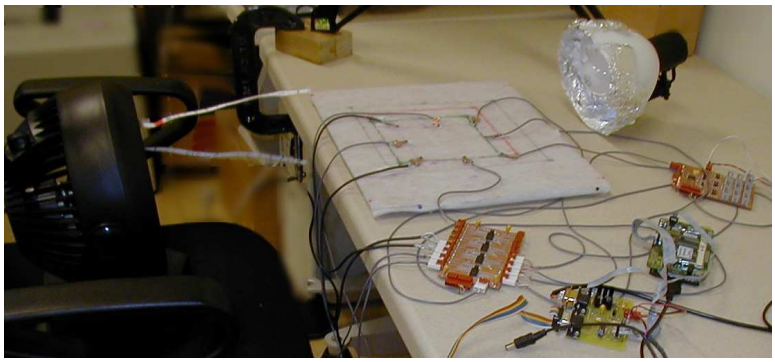


Figure 4.17. Photograph of the turbulent implementation setup for the gradient source localizer. An oscillating fan (rotating horizontally from 45 to 135 degrees on it's axis) is placed 20 **cm away from the sensor array.**

simple case in Fig. 4.19. In Fig. 4.19 (a) The Form 1 $\mathbf{A}_{\text{init}}$ and the Form 3 $\mathbf{A}_{\text{init}}$ again are too slow to converge while the Form 2 $\mathbf{A}_{\text{init}}$ quickly tracks the mean. In Fig. 4.19 (b), the best tracker is the uniform 5-banded (Form 2, $S_c = 5$) $\mathbf{A}_{\text{init}}$.

4.11.3 Turbulent and Noisy Environment Results

Again, an oscillating fan was placed at a $-90°$ angle, 20cm from the board. In addition, sensor noise of $+/-2°C$ was added to each sensor measurement. In Fig. 4.20, variation due to turbulence is seen at the beginning, but Gaussian noise is the predominant sensor disturbance as the angle converges. The simple algorithm is highly sensitive to noise. In Fig. 4.20 (a), the Form 1 $\mathbf{A}_{\text{init}}$ and the Form 3 $\mathbf{A}_{\text{init}}$ do a good job, but again the Form 2 $\mathbf{A}$ is the best tracker of the true angle. In Fig. 4.20 (b), the performance of both the $\mathbf{A}_{\text{init}}$s of Form 2 and Form 3 cooperation are similar. All algorithms are robust to heavy, additive Gaussian noise.

A summary of the implementation results can be seen in Table 4.5.

4.12 Steady-State Analysis of Stationary Array

To simulate the steady-state case, a constant vector with added noise was continually input into the sensor array. The noise level standard deviation is 10% of the norm of the input vector (i.e. $\tilde{2}0$ dB). The algorithm is the same as Section 4.6, but the input vector, v, is kept

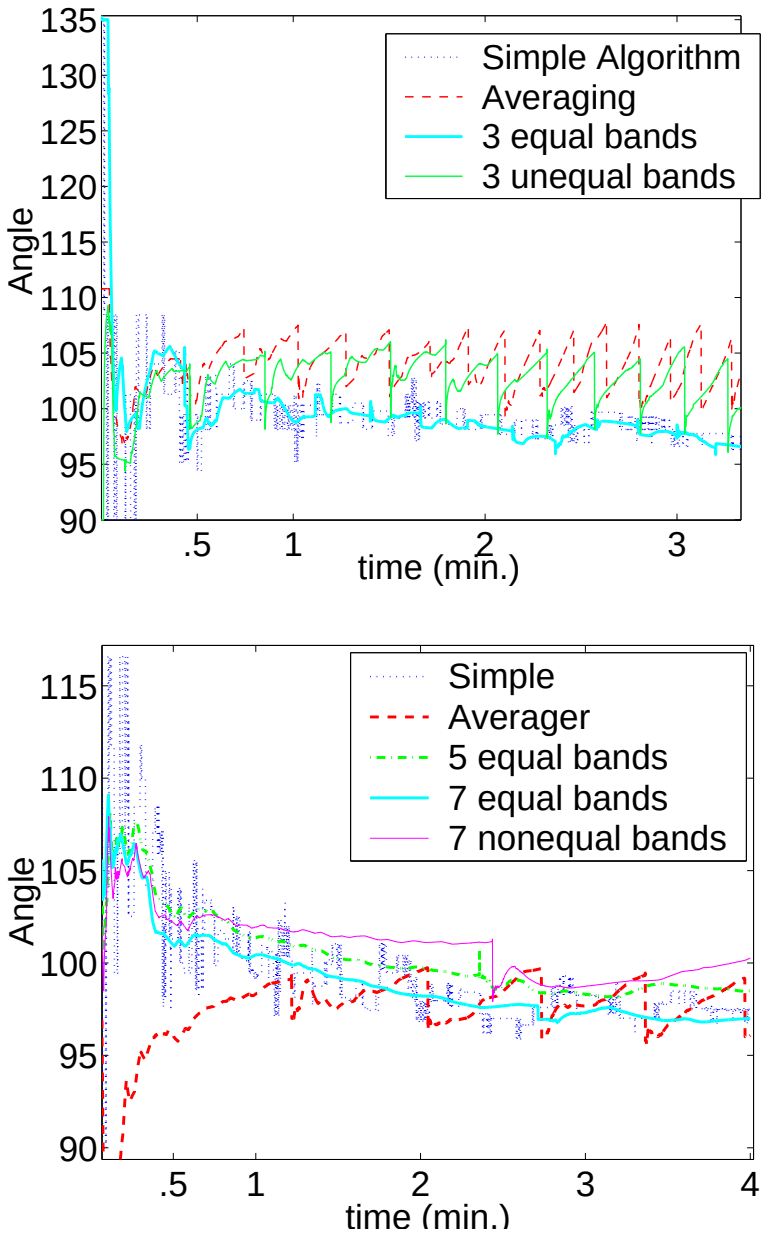


Figure 4.18. Diffusive environment. a) 4**-sensor localization b)** 8**-sensor localization of** 100° **source in a diffusive environment. The simple algorithm determines the angle using just the input,** $\mathbf{v}[n]$**, without memory. The averager is when** $\mathbf{A}_{\text{init}}$ **is of Form 1, the** S_c **equal bands represent Form 2** $\mathbf{A}_{\text{init}}$**, and the unequal** S_c**-bands represent Form 3** $\mathbf{A}_{\text{init}}$**. Note that the uniform** $N-1$ **-banded sensor cooperation performs the best.**

constant for every iteration of the algorithm.

In Figs. 4.21, 4.22, and 4.23, various angles are simulated and convergence of the different forms of $\mathbf{A}_{\text{init}}$ are varied. Also, a case is examined which goes beyond the odd-banded case – having a full $\mathbf{A}$ matrix instead of banding-limiting it. For example, the $N = 4$,

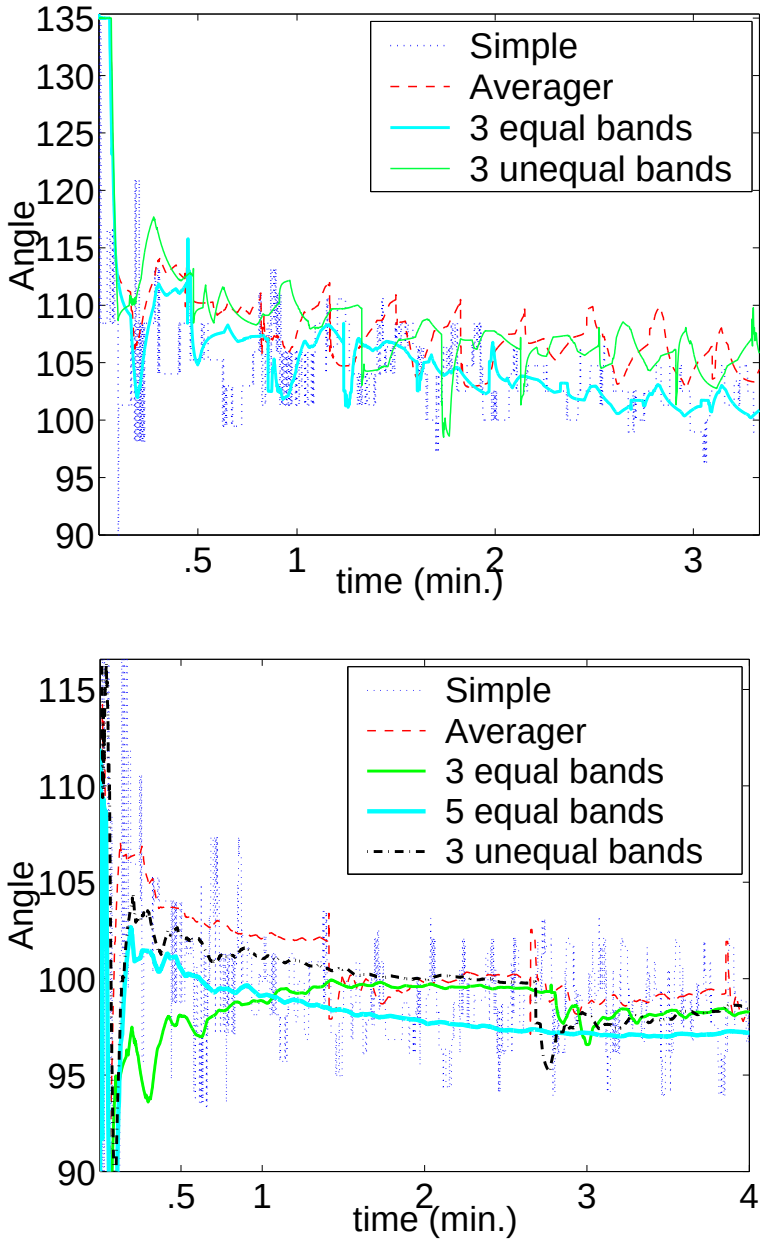


Figure 4.19. Turbulent environment. a) 4 **sensor localization b)** 8 **sensor localization of** 100° **angle in turbulent environment. Note that the Form 2,** $S_c = 3$ $\mathbf{A}_{\text{init}}$ **performs the best in the** 4**-sensor case, and the Form 2,** $S_c = 5$ $\mathbf{A}_{\text{init}}$ **performs best in the** 8**-sensor case.**

$S_c = 4$ case has no sensor cooperation limits on the matrix so the matrix is an unrestricted 4×4 matrix. This shows "full" sensor cooperation without any limitations.

In Fig. 4.21, the full-matrix case has the largest noise variance. While the $S_c = 1$ case quickly converges to a steady-state of 45 degrees, the $S_c = 3$ case slowly converges to the

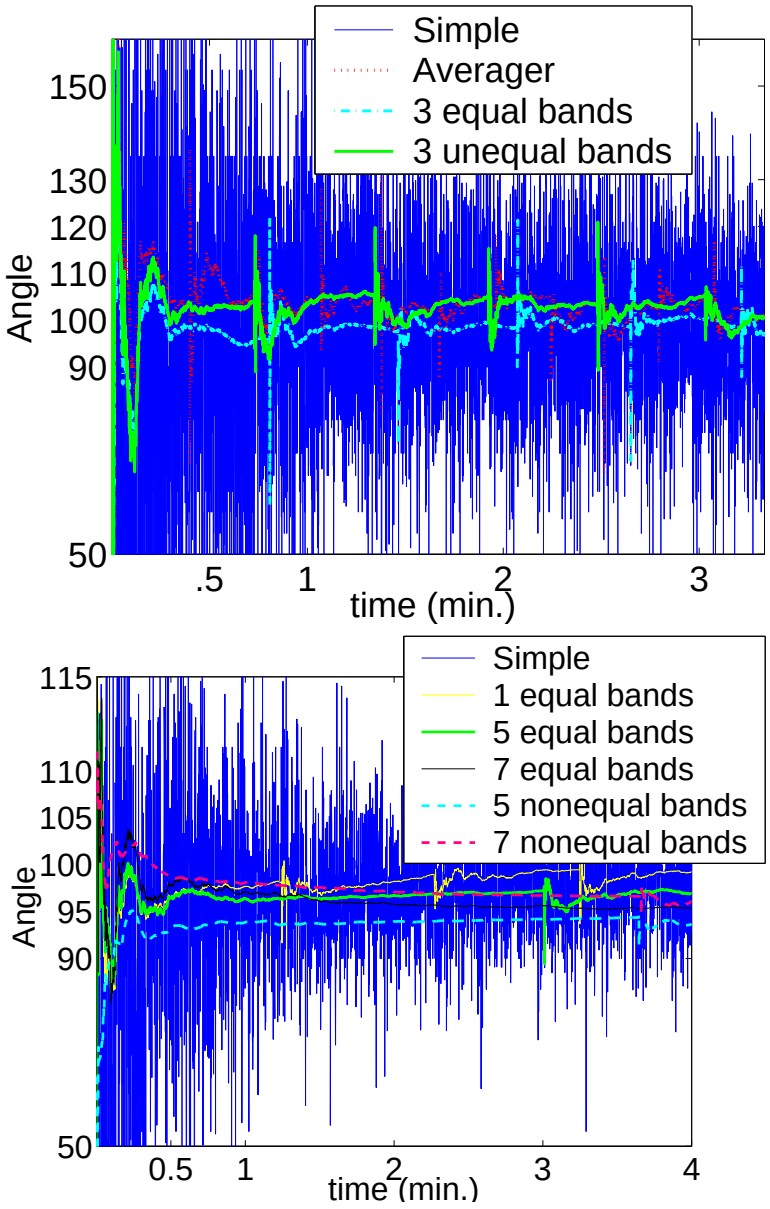


Figure 4.20. Turbulent environment with additive $+/-2^\circ$ Celsius on the sensor measurements. a) 4**-sensor localization b)** 8**-sensor localization of** 100° **angle in a turbulent environment. Note the Form 2,** $S_c = N - 1$ $\mathbf{A}_{\text{init}}$ **tracks the best angle, but all algorithms yield similar results.**

true angle and is not as effected by the noise variance. So, for the static, steady-state case, the algorithm has a trade-off between biasing the angle localization and minimizing the the noise variance.

In Fig. 4.22, it is thought that the tangent function reduces the additive noise; the hypothesis is that the closer the true angle of localization is to 45°, the less sensitive the

Table 4.5. Performance summary of a 4-sensor array 100°-angle source localization in different environments using the last minute of data collected. The deviation of the mean angle from the 100° and the standard deviation of the last minute of data are shown. The Form 2 A_{init} sensor cooperation clearly reduces the standard deviation of the angle estimate while more accurately tracking the mean as opposed to the other methods.

	Diffusive		Turbulent		Turbulent + Noise	
	△ Mean	Std	△ Mean	Std	△ Mean	Std
Simple (1)	−0.2°	0.9°	2.2°	2.0°	−1.3°	18.2°
Averager (2)	3.7°	2.1°	5.8°	1.89°	2.0°	2.8°
BFE3 (3)	−2.2°	0.7°	1.6°	0.8°	−0.5°	1.8°
BFT3 (4)	2.1°	1.8°	5.7°	1.44°	2.1°	2.3°

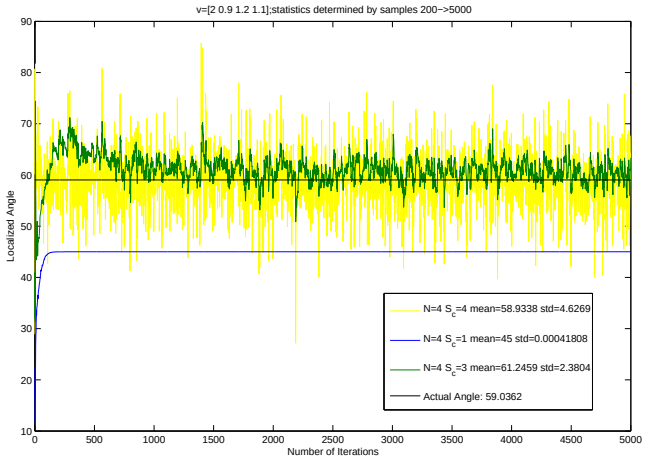


Figure 4.21. The input vector, v = [2 0.9 1.2 1.1] (59° source angle), + noise for a 4 sensor array using different levels of sensor cooperation. $S_c = 4$ represents the full A matrix with no sensor cooperation constraints. Mean and STD are determined from 200 → 5000 samples.

tangent function is to the noise. In addition to the $S_c = 1$ case which quickly converges to a steady-state of 45 degrees, the $S_c = 3$ is skewed from the true angle as well. The full matrix again has the best localization angle but is a little more sensitive to noise.

In Fig. 4.23, the tangent function can also cause wide instabilities when the function is converging to the true angle as seen in the green curve. While the $S_c = 1$ case quickly converges to a steady-state of 45 degrees, the $S_c = 3$ case converges to a skewed angle but not as skewed as the $S_c = 1$ case, and it is not as effected by the noise variance as the full matrix.

In Fig. 4.24, both the $S_c = 1$ and $S_c = 3$ case are skewed and have similar variance to

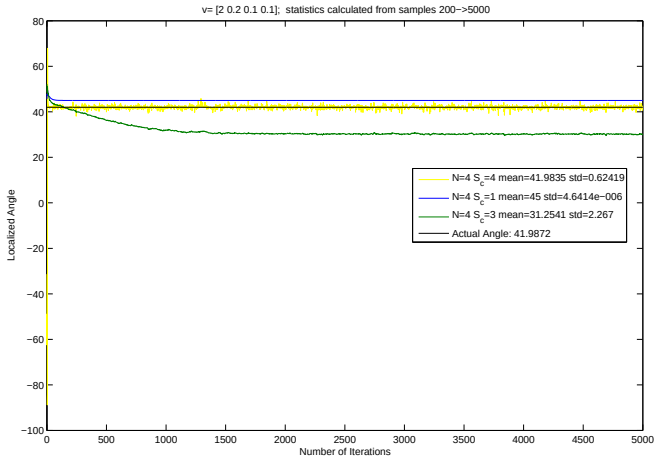


Figure 4.22. The input vector, $v = [2\ 0.2\ 0.1\ 0.1]$ **(** 42° **source angle), + noise for a 4 sensor array using different levels of sensor cooperation.** $S_c = 4$ **represents the full A matrix with no sensor cooperation constraints. Mean and StD (standard deviation) are determined from** $200 \rightarrow 5000$ **samples.**

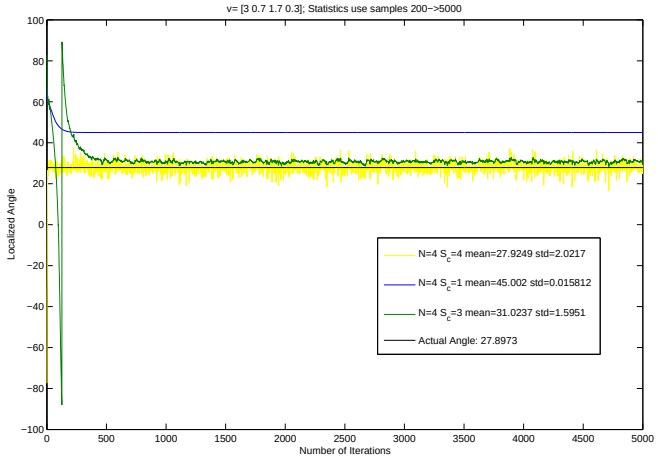


Figure 4.23. The input vector, $v = [3\ 0.7\ 1.7\ 0.3]$ **(** 28° **source angle), + noise for a 4 sensor array using different levels of sensor cooperation.** $S_c = 4$ **represents the full A matrix with no sensor cooperation constraints. Mean and STD are determined from** $200 \rightarrow 5000$ **samples.**

the untapered case. There is no advantage for the stationary case to use the tapered version of the algorithm in this case.

Examples for an 8-sensor array are shown in Figures 4.25 and 4.26. In Fig. 4.25, sometimes the sensor cooperation does as well as the full-matrix case and with lower variance. But in some cases, as shown in Fig. 4.26, the sensor cooperation localizes the angle better than no sensor cooperation, but not as good as the full-matrix case. The sensor cooperation

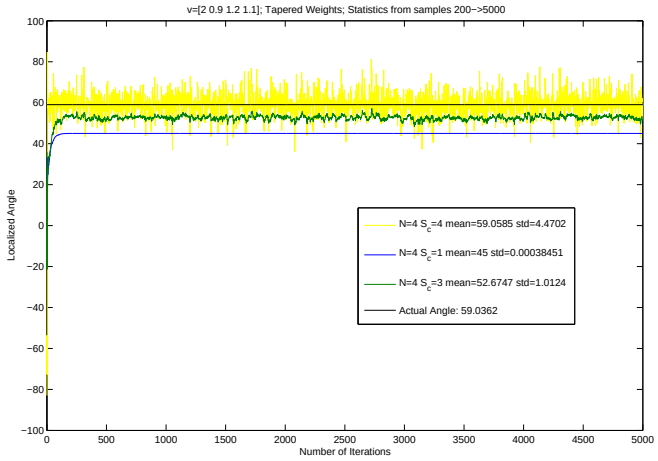


Figure 4.24. The input vector, $v = [2\ 0.9\ 1.2\ 1.1]$ **(** $59°$ **source angle), + noise for a 4 sensor array using different levels of TAPERED sensor cooperation.** $S_c = 4$ **represents the full (non-tapered) A matrix with no sensor cooperation constraints. Mean and STD are determined from** $200 \rightarrow 5000$ **samples.**

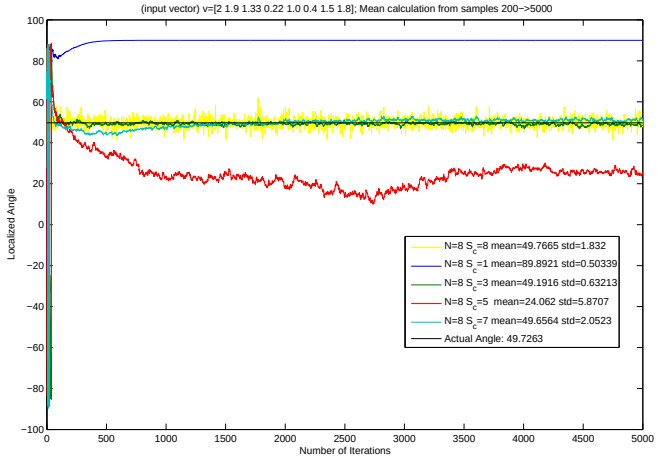


Figure 4.25. $v = [2\ 1.9\ 1.33\ 0.22\ 1.0\ 0.4\ 1.5\ 1.8]$**,** $50°$ **source angle + noise for an 8-sensor array using different levels of sensor cooperation.** $S_c = 8$ **represents the full A matrix with no sensor cooperation constraints. Mean and STD are determined from** $200 \rightarrow 5000$ **samples.**

usually introduces equal or lower variance compared to the full-matrix case.

4.13 Conclusions

Current implementations to track heat and chemicals are underdeveloped, complicated, and/or costly. For a cost-effective solution, we propose a small sensor array enhanced by a chemotaxis-inspired, Hebbian learning algorithm. Bacterial membrane cell receptors are

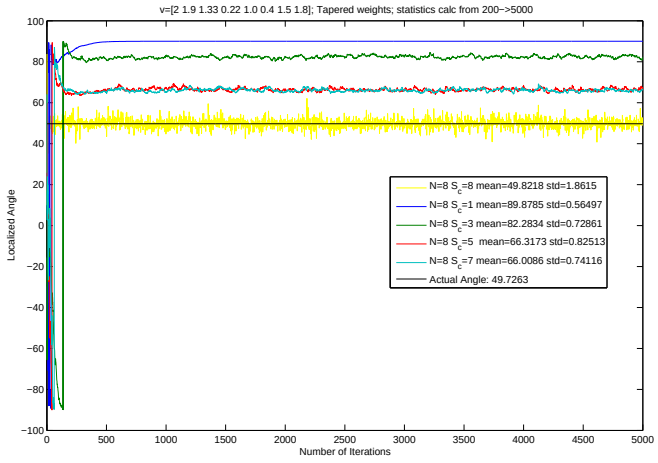


Figure 4.26. v = [2 1.9 1.33 0.22 1.0 0.4 1.5 1.8]**,** 50° **source angle + noise for an 8-sensor array using different levels of TAPERED sensor cooperation.** $S_c = 8$ **represents the full A matrix with no sensor cooperation constraints. Mean and STD are determined from** $200 \rightarrow 5000$ **samples.**

approximated with a square array for implementation, and the sensor array incorporates various types of sensor cooperation into the adaptive Hebbian algorithm as it tracks the source.

Turbulence and noise play a major role in implementation due to the need to track light traces of chemicals in an environment. Simulations of a mobile array are run in various noisy conditions for three different sensor cooperation constraints: Form 1 (no sensor cooperation, classical sensor averaging), Form 2 (full-sensor cooperation), and Form 3 (a side-sensor attenuation). We show sensor cooperation generally improves source localization time over the classical averaging. The Form 2 constraint sacrifices a little performance in high SNR for significantly improved performance in low SNR, while the Form 3 constraint yields a consistent incremental improvement in all SNRs.

We also show that sensor cooperation helps real-time Hebbian weight adaptation of a stationary implementation in diffusive, turbulent, and noisy environments, thus speeding up convergence time and minimizing standard deviation when localizing a source. For the stationary case, the Form 2 constraint localizes the source more accurately by reducing the standard deviation of the estimate by 2-fold over the Form 1 and Form 3 constraints. On the

other hand, the sensor cooperation algorithm biases the convergence angle when the input is a constant angle vector in with noise, thus it is suboptimal. Some behavior like this was seen in the real-world implementation, but perhaps this biasing was not always witnessed in only 2 minutes of the data. In conclusion, the groundwork is laid for the gradient source localizer hardware, and performance is improved using a modified Hebbian algorithm that exploits the sensor geometry. The method is simple enough to be implemented in low-power analog circuitry in the future, allowing its use in ubiquitous applications.

CHAPTER 5

TURBULENT PLUME ANALYSIS

So far, in our chemical/temperature measurement scenarios, we have focussed on the pure diffusive environment, modelled at time to infinity with a $\frac{1}{r}$ diffusion in space. While this scenario is an interesting case when we have a very slowly decaying amplitude over the closely spaced sensors, this scenario is never likely in the real-world for chemical diffusion. It is a likely scenario for temperature diffusion as shown in our implementation. We now want to model turbulent chemical scenarios, but there are many factors to consider due to the nonlinear nature of turbulence. Chemical plume turbulence is quasi-periodic and can be modeled with a linear model depending on the **Reynolds number**. The Reynolds number is the ratio of inertial forces to viscous forces and is used for determining whether a flow will be laminar or turbulent [6]. At various Reynolds numbers, one can have periodic wakes/eddies (on the boundary), called Vortex Karman Streets in our case, and at high Reynolds numbers, unsteady vortices appear on many scales and interact with each other.

Due to the unsteady nature of turbulence, we elected to receive controlled turbulent plume measurements rather than generating a computational model. In this thesis, we will discuss our considerations of this turbulent data and our analysis.

5.1 Data Collection and Noise Analysis

The data we analyze was taken by Donald Webster's lab in the School of Civil and Environmental Engineering at Georgia Tech. His goal was to create a controlled environment to produce examples of vortex shedding.

5.1.1 Karman Vortex Principles

Prof. Dr. Chiang Shih of Florida State University has eloquently described vortex shedding and the relationship between the diameter of the cylinder in the path of a flow, the velocity of the flow, and the vortex shedding frequency: "Vortex Shedding: The boundary

layer separates from the surface forms a free shear layer and is highly unstable. This shear layer will eventually roll into a discrete vortex and detach from the surface (a phenomenon called vortex shedding). Another type of flow instability emerges as the shear layer vortices shed from both the top and bottom surfaces interact with one another. They shed alternatively from the cylinder and generates a regular vortex pattern (the Karaman vortex street) in the wake. The vortex shedding occurs at a discrete frequency and is a function of the Reynolds number. The dimensionless frequency of the vortex shedding, the shedding Strouhal number,

$$S_t = \frac{f_s D}{V} \tag{5.1}$$

, is approximately equal to 0.21 when the Reynolds number is greater than 1,000. F_s is the vortex shedding frequency, D is the diameter of the cylinder, and V is the inflow velocity. ". The surface he is referring to is a cylindrical surface, obstruction in the pathway of the flow. Taneda took photos of the following effects of the Rayleigh number on Karman vortex streets [100] :

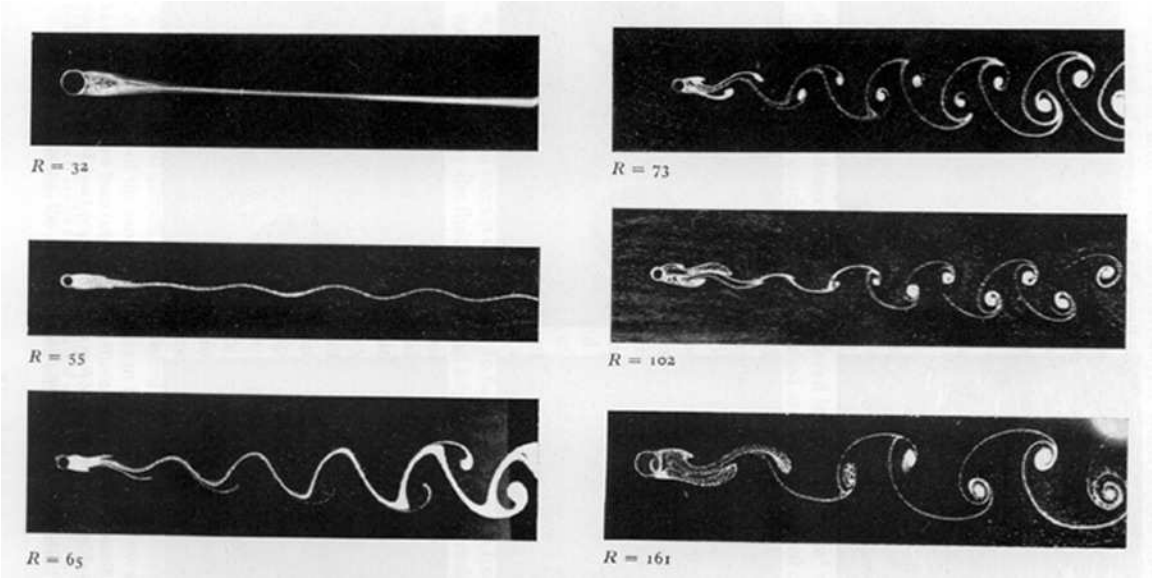


Figure 5.1. The effect of the Rayleigh number on cylindrical Karman vortex streets.

5.1.2 Planar laser-induced flourescence data

From [45]:

"Planar laser-induced fluorescence (PLIF) is a non-intrusive, optical measurement technique to obtain a sequence of instantaneous, high-resolution spatial concentration fields. Examples of the plumes taken with this method can be seen in Figs. 5.31 and 5.32. The data collected for plumes in a fully developed open channel flow was used for the spectral analysis. The flow was established in a 1.07 m wide, 24.4 m long tilting flume with rectangular cross-section and smooth bed. The average velocity in the flume was 5.0 cm/s and the flow depth was 20.0 cm. A small amount of fluorescent dye, Rhodamine 6G, was mixed with the source effluent such that the plume contained extremely low dye concentrations, of the order of 10 μ g/l. The effluent velocity was matched with the channel flow velocity thus creating a passive source and avoiding the production of additional turbulence by shear induced by the effluent. Sweeping an argon-ion laser beam in a plane parallel to the bed with a scanning mirror created the illumination sheet. The laser light caused the dye to fluoresce and a digital CCD camera (8bit gray scale, with 1018 vertical and 1008 horizontal pixels) captured the emitted light. The light intensity emitted by the dye is directly proportional to the dye concentration and laser intensity. However, the obtained raw images suffer from laser sheet non-uniformity, lens vignette and pixel variability. Therefore, an in situ calibration was performed to convert the raw images into quantitative data of concentration field. The sweep duration of the laser was shorter than any time scales in the flow, thus the images were truly frozen in time. For the data presented here, 6000 images were captured with 10 frames/s. The field of view was 1m $\times$ 1m and, therefore, the spatial resolution was roughly 1mm. The laser sheet was in the same horizontal plane as the plume source, 2.54cm above the floor. The obtained data represents the two-dimensional concentration field at this elevation."

We wished to reduce the images to the most active regions therefore, the 1024×1024 image was cropped to a 401×940 pixel image. In the vertical direction, pixels $251 \rightarrow 651$

were taken, and in the horizontal direction, pixels 76 → 1015 were taken.

From this point, when we refer to the coordinates of the image, they are the coordinates of the cropped image. (0,0) is the upper-right hand corner of the image. The first coordinate is how many rows DOWN and the second coordinate is how many rows ACROSS.

5.1.3 Noise and Sensitivity Analysis

The CCD camera introduces noise into the plume image. To measure these effects, we took four corners of the image as seen in Fig. 5.2.

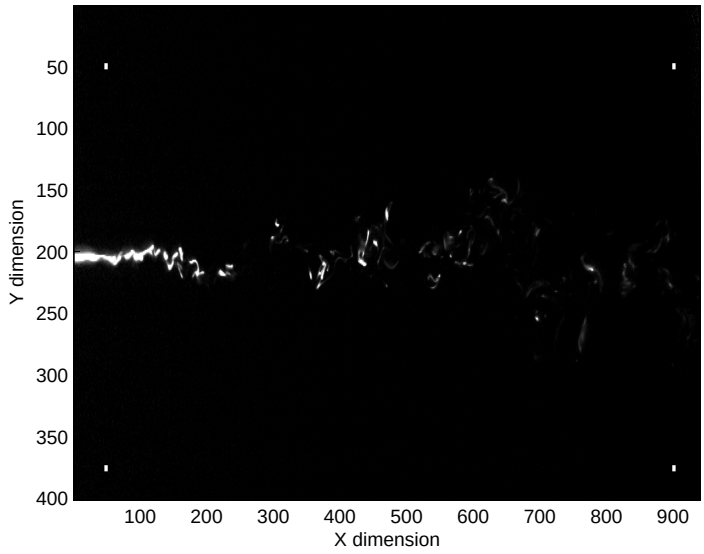


Figure 5.2. Four arrays of 25 sensors are seen near the corners of the cropped image.

The 600 second time series from the 25 sensors are spatially averaged to show the effect of noise seen in Figs. 5.3 through 5.6.

From these figures, we can conclude that there is a slight diffusion effect. (Histograms not shown here were visually-inspected to give a rough idea of the precision of the data). The sensor arrays nearer to the source, shown in Figs. 5.3 and 5.4, are exactly on the threshold of detection. Almost all pixels are barely registering above the 0, black, value. The sensors exhibit a Gaussian distributed error on this value. On the other hand, the

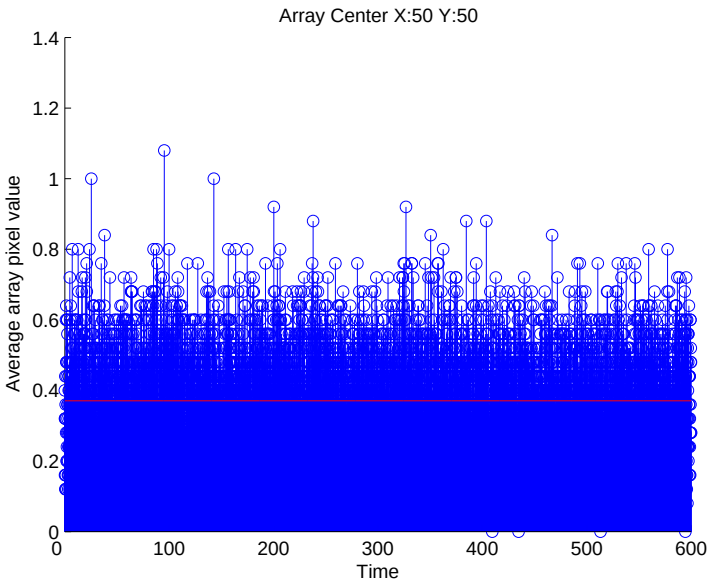


Figure 5.3. The spatially-averaged time profile of the array at the $(50, 50)$ **position (upper left hand array).**

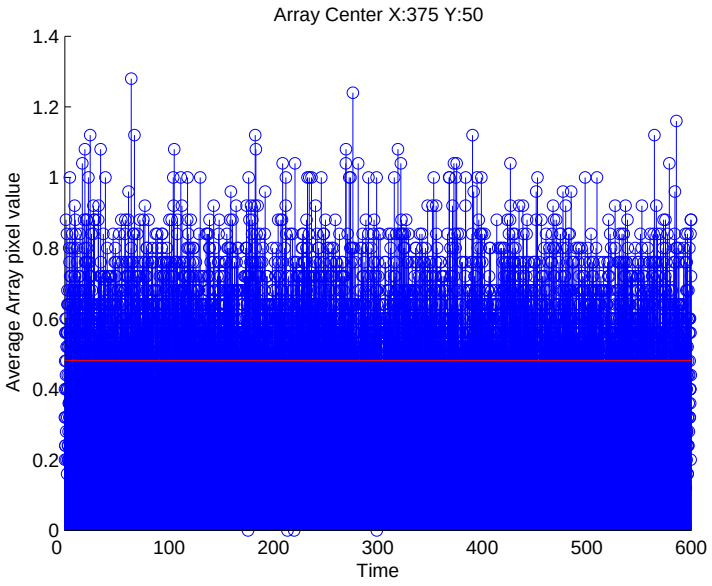


Figure 5.4. The spatially-averaged time profile of the array at the $(375, 50)$ **position (lower left hand array).**

sensors furthest away from the source, shown in Figs. 5.6 and 5.5, exhibit more of shot-noise with a Poisson distribution, and the computed mean is a lot lower. So, we conjecture when the signal is almost non-existent, the camera still has the occasional shot noise, but on detectable signals, the noise introduced by the camera is Gaussian-like.

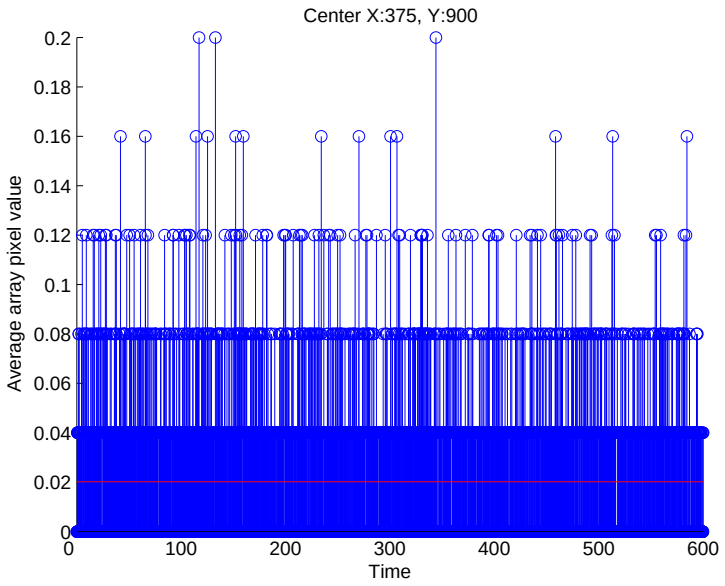


Figure 5.5. The spatially-averaged time profile of the array at the $(375, 900)$ **position (lower right hand array).**

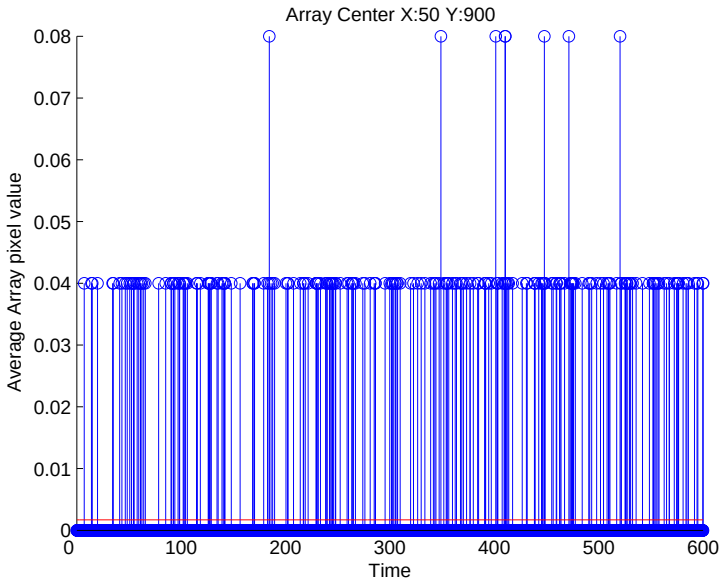


Figure 5.6. The spatially-averaged time profile of the array at the $(50, 900)$ **position (upper right hand array).**

5.2 Spectral Analysis

The focus in [45] is that the vortex shedding that occurs is spectrally encoded into the plume. Weissburg/Janata et al. [107] have hypothesized that animals such as lobsters use such frequency information to help it localize a food source.

This concept needs to be more thoroughly explored. Such phenomenon can also be found with human hearing, the brain not only uses interaural intensity difference (IID) but inter-aural time delay (ITD), to localize a sound source. The interaural time delay is most important when low frequencies, with wavelengths equal to or greater than your head width, pass laterally through your head and create a detectable phase difference. The interaural intensity difference results at high frequencies when the signal is too fast for phase to be detectable. If such a concept is to be claimed to be true in biological plume detection, IID and ITD should be explored in this regime. We explore these aspects in Section 5.3.1.

5.2.1 The Magnitude FFT/Power Spectrum measurements of the plume data

Since there is a vortex shedding frequency when the signal is obstructed with a cylinder, we decided to go about measuring such a frequency. In the modulated experimental data, the flow velocity was $U = 5.0\text{cm/s}$ and the diameter of the cylinder is 0.8cm yielding an approximate 1.3Hz modulation, given in (5.1).

Let us first examine the modulated plume then compare it to the plume without modulation. First, a straight-8192 point FFT of the 600-second data record was taken (see Fig.

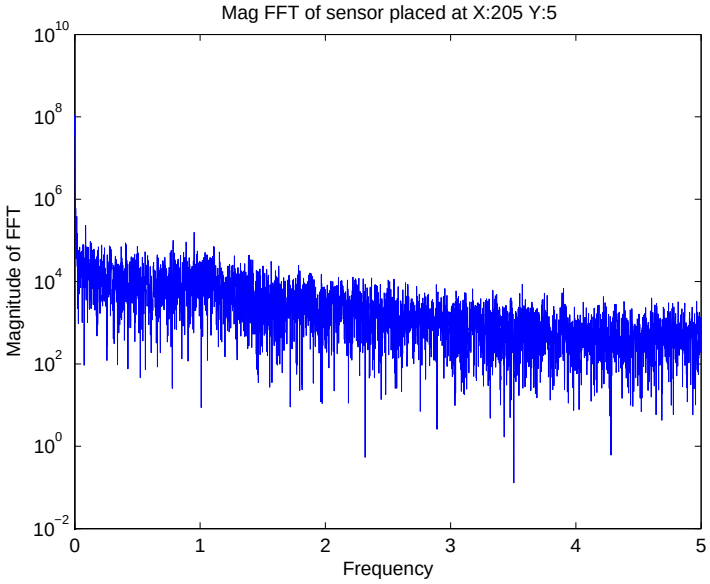


Figure 5.7. 600 second magnitude FFT of modulated plume data with a sensor placed at (205,5).

5.7). This did not show a clear a 1-Hz periodicity. Next, to get a smoother spectrum, we made averaged Power Spectral Density plots, using 100 windows of 6 seconds duration to get the following plots. The FFT size used was 512. First, a rectangular window was used in the spectrum estimation in Fig. 5.8. Then a Hann window was tried (see Fig. 5.9).

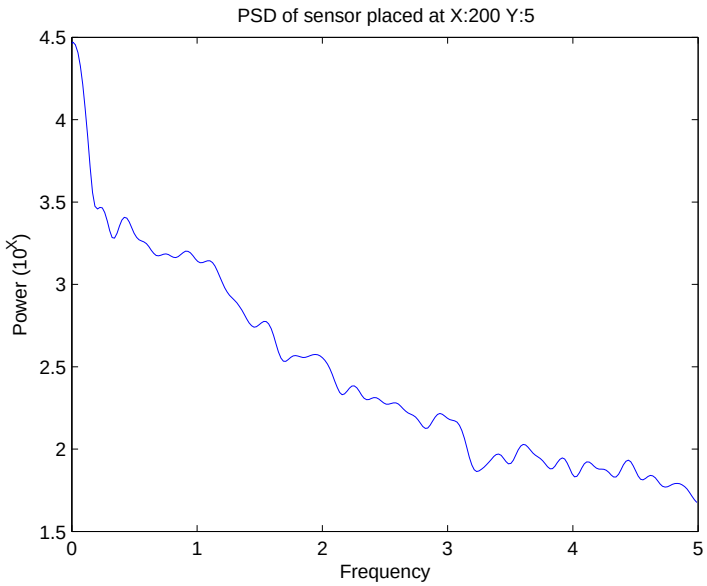


Figure 5.8. 100 averaged spectrums using a square window with a sensor placed at (200,5).

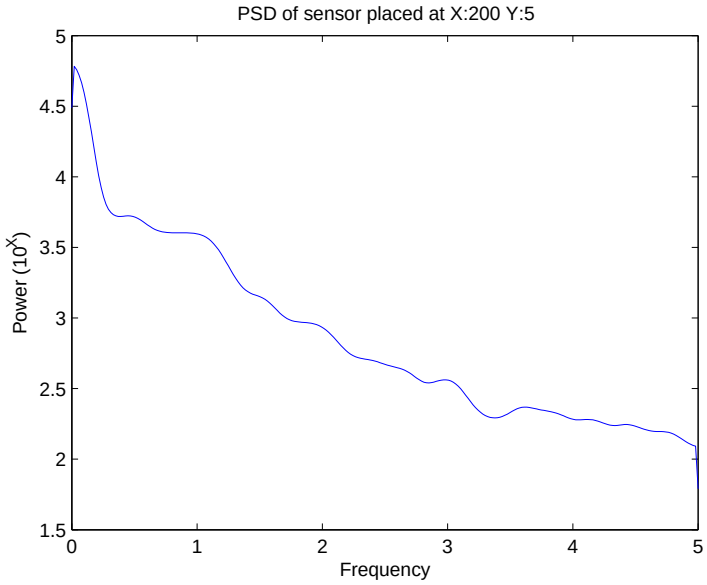


Figure 5.9. 100 Averaged spectrums using a Hann window with a sensor placed at (200,5).

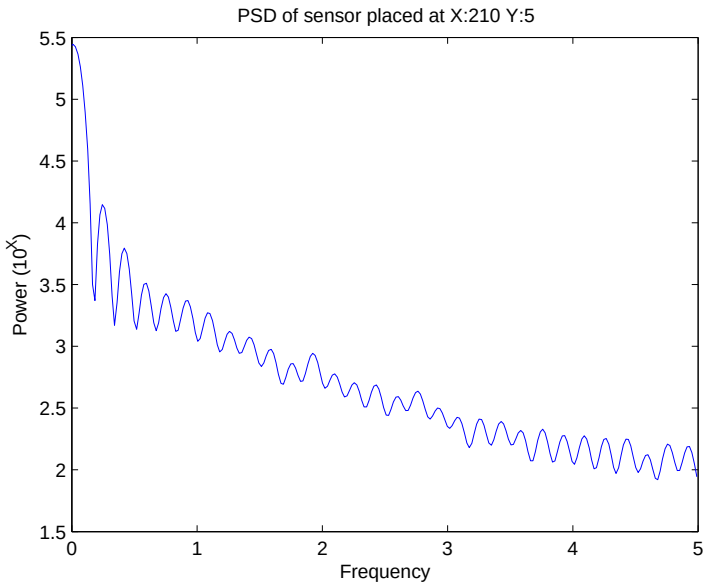


Figure 5.10. 100 Averaged spectrums using a square window with a sensor placed at (210,5).

As we can see from the plots, the Hann window smooths the line even further and gets rid of the sidelobe effect of the rectangular window. An amplified rectangular window can be seen in Fig. 5.10.

Then, we took a sampling of three rows to get an idea of how the centerline vs. off-centerline data will look. Also, we took a sampling of three columns to see how distance from the source affects the spectrum. The values of the rows were 200, 205, 210, 205 being the visually determined centerline. The column values were 5, 50, 500.

The comparison of varying the sensor position on the horizontal level can be seen in Fig. 5.11. The interesting thing here is that 5 pixels away from the source (equivalent to 5mm), we see a slight periodicity at 1 Hz for 200 and 205 (stronger for the 205^{th} row, the true centerline). Yet this periodicity is not that strong.

The comparison of varying the sensor position from the distance away from the source can be seen in Fig. 5.12. We chose the centerline that did the best in the previous comparison, and we varied the distance of the source. In this case, the 1 Hz signal completely disappears even for only 5cm away from the source. But with this illustration, the decay of the total power of signal is seen and almost looks linear.

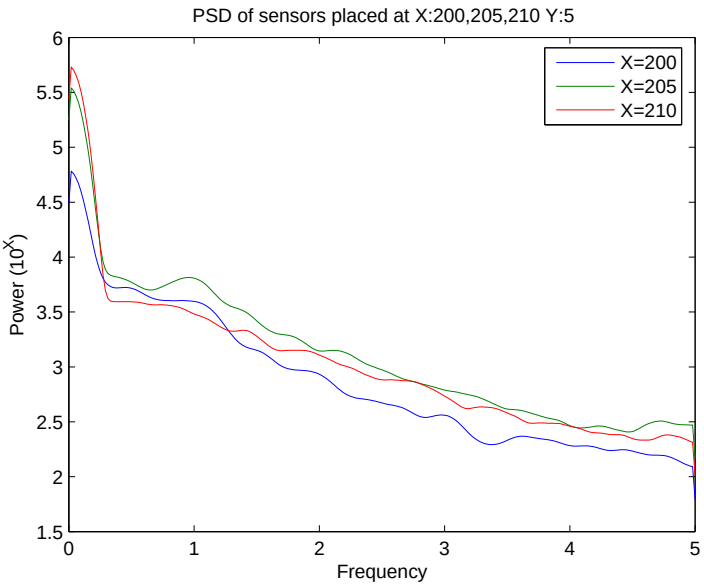


Figure 5.11. Comparison of spectrums at different "centerlines" in the plume.

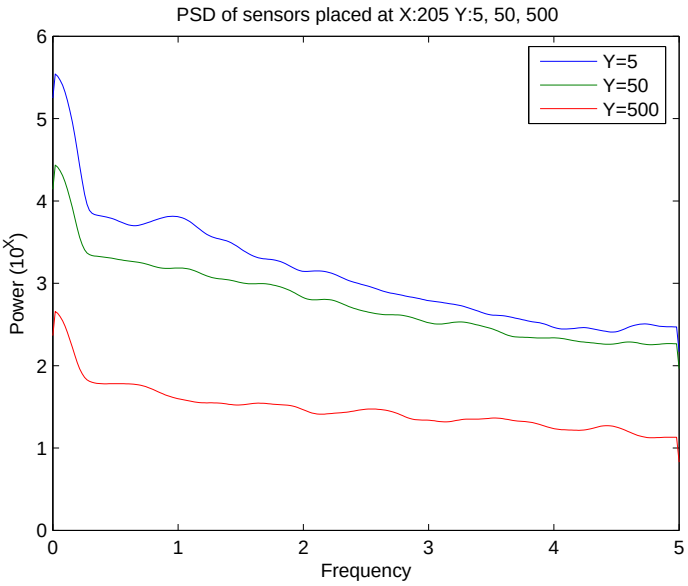


Figure 5.12. Comparison of spectrums at different distances away from the source in the plume.

5.2.2 Coherency

We will use this section to mainly review the work of [45]. They focused upon cross-coherence as a way to gain spectral information about the turbulent field. Their setup of sensors can be seen in Fig. 5.13. Here, we will develop the mathematical framework needed [98].

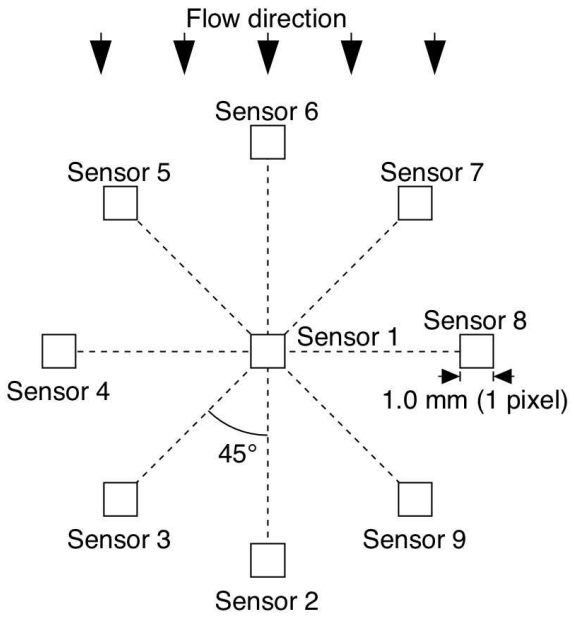


Figure 5.13. Hiroshi/Janata's sensor setup

Let us first define the auto-correlation sequence, $r_{yy}(k)$ as:

$$r(k) = E[y(t)y^*(t-k)] \tag{5.2}$$

and the power spectral density, $\phi(\omega)$,

$$\phi(\omega) = \sum_{k=-\infty}^{\infty} r(k)e^{-jwk\omega} \tag{5.3}$$

The cross-correlation sequence can be subsequently defined as:

$$c_{yu}(k) = E[y(t)u^*(t-k)] \tag{5.4}$$

, and the cross-correlation power spectral density, $\phi_{yu}(\omega)$, as:

$$\phi_{yu}(\omega) = \sum_{k=-\infty}^{\infty} c_{yu}(k)e^{-jwk\omega} \tag{5.5}$$

.

The cross spectrum, or complex-coherency, of two signals, $y(t)$ and $u(t)$ is defined as:

$$C_{yu}(\omega) = \frac{\phi_{yu}(\omega)}{[\phi_{yy}(\omega)\phi_{uu}(\omega)]^{1/2}} \tag{5.6}$$

Hiroshi et. al use $|C_{yu}(\omega)|$, the coherency spectrum. It can also be viewed as the correlation-coefficient between the signal components vs. frequency.

One more note is the result of the Wiener-Hopf equation. If two signals are linearly related,

$$y(t) = \sum_{k=-\infty}^{\infty} h_k u(t-k) \tag{5.7}$$

one can take the minimum of the mean square error function, $E[(y(t) - \sum_{k=-\infty}^{\infty} h_k u(t-k))^2]$, with respect to the filter. In such a case, the filter is equivalent to the cross-spectrum of the two signals divided by the auto-spectrum of the input signal.

$$H(\omega) = \frac{\phi_{yu}(\omega)}{\phi_{uu}(\omega)} \tag{5.8}$$

If they are linearly related with error, the error will be

$$\begin{aligned} e(t) &= y(t) - \sum_{k=-\infty}^{\infty} h_k u(t-k) \\ E(\omega) &= Y(\omega) - H(\omega)U(\omega) \end{aligned} \tag{5.9}$$

Parseval's Theorem states that:

$$E[(e(t))^2] = E[(E(\omega))^2]$$

and

$$E[(E(\omega))^2] = E[(Y^2(\omega) - 2H(\omega)U(\omega)Y(\omega) + H^2(\omega)U^2(\omega))].$$

.

Substituting into (5.8), we get $\phi_{yy}(\omega) - 2\frac{\phi_{yu}(\omega)}{\phi_{uu}(\omega)}\phi_{yu} + \frac{(\phi_{yu}(\omega))^2}{(\phi_{uu}(\omega))^2}\phi_{uu}(\omega)$ and using the definition from (5.6) results in:

$$E[(E(\omega))^2] = (1 - |C_{yu}(\omega)|^2)\phi_{yy} \tag{5.10}$$

In (5.10), if the magnitude coherency spectrum is equal to one, then we can say that (5.7) holds true, and the input and output signals are linearly related. We have presented this partial derivation of the expectation of the error signal in order to set a later discussion in Section 5.4.

5.2.3 MSCohere Results

In this section, we try to reproduce Hiroshi's results where the magnitude of (5.6) is used to evaluate the plume. But, if we take too long a data record (we have 600 seconds of data), then the results may be too noisy. In [45], Hiroshi compares averaged coherence plots using 100 records of length 600 (60 seconds) and 50 records of length 1200 (120 seconds each). We verified his result that averging 100 records has less noise and seems to yield some significant peaks in the spectrum. What needs to be verified in the future is that there is not a more optimal resolution vs. noise averaging trade-off. We obtained reasonable plots with the 100 averaged records of 60 seconds, so we accepted this measure on faith.

In [45], the author found that there is a very high coherence at 1 Hz in the modulated plume. If we wish to incorporate sensor array signal processing methods, we would like to assume there is a signal existing at one frequency. From experiments in Section 5.2.1, we did not find an explicit 1 Hz signal in the plume. Hiroshi found a significant coherence between sensors at this frequency, so we have continued along these lines to verify the results.

In Fig. 5.14, we examine what a coherence spectrum looks like for each sensor orientation. This is similar to determining the coherence of each spatial orientation in the plume. The sensors perpendicular to the plume exhibit the strongest correlation for almost all frequencies, but this is understandable because the same events will reach these sensors at the same times. Also, sensors 1 and 3 which are at a -45° angle and sensors 1 and 5 which are at a -135° angle exhibit a relatively high correlation at the 1.3 Hz frequency, but they also are strong at 0.4 Hz. So, to say these signals are just correlated at 1 Hz may not be consistent with all sensor placements.

In Fig. 5.15, we examine the coherence spectrum of the perpendicular sensors, 1 and 4, at various distances. The interesting thing here is the coherence at 0.47 Hz increases with distance from 5 to 7 cm. And for 8 to 10 cm, the coherence seems to shift to 0.66 Hz. At 5 to 8 cm, there seems to be a slight correlation at around 1 Hz. The 7 and 8 cm distances

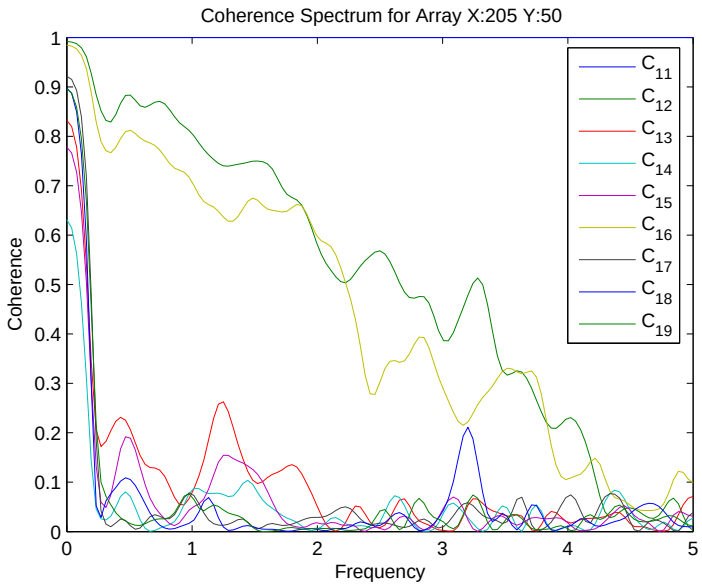


Figure 5.14. The coherence spectrum of an array placed at (205,50) with 1 cm between each sensor.

almost exhibit coherence at harmonic frequencies after that.

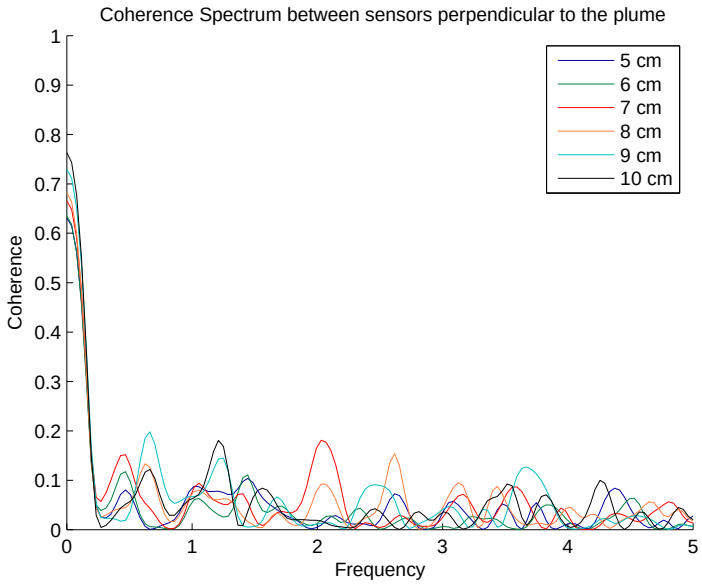


Figure 5.15. Comparison of coherence of sensors 1 and 4 of an array at (205,50:10:100), short range distances.

While coherence information is contained in the sensors perpendicular to the plume, little information exists in the coherence spectrum of the sensors parallel to the plume. This can be seen in Fig. 5.16. We will examine sensor orientation further in Section 5.3.

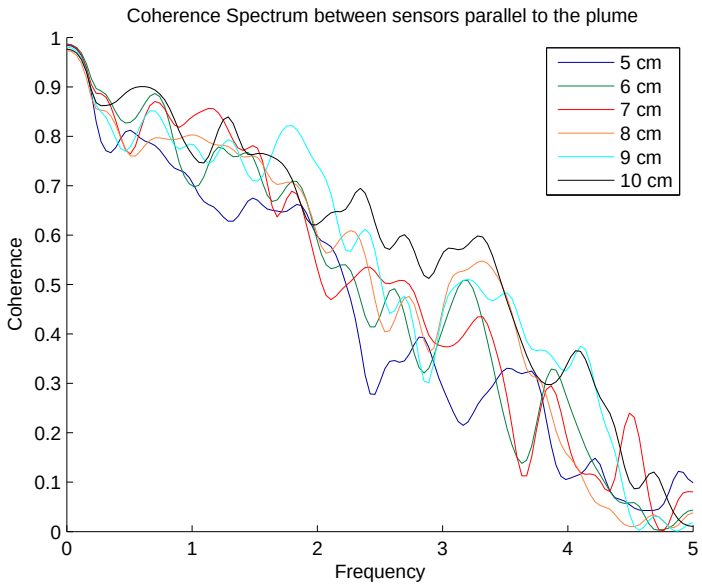


Figure 5.16. Comparison of coherence of sensors 1 and 4 of an array at (205,50:10:100), short range distances.

In Fig. 5.17, we examine the coherence spectrum of the perpendicular sensors, 1 and 4, at various long range distances. The coherence peaks seem to be almost separated with an equidistance of 0.15 Hz. But, it is not linear, sensors 1 and 4 placed at 70 cm have a peak at 0.55 Hz while when the array is placed at 10 cm, the sensors have a peak at 0.67 Hz and 1.21 Hz. This graph clearly shows that the modulation frequency shifts around with the array placement in the plume.

In Fig. 5.18, we examine the coherence spectrum between sensors when there is only 0.5 cm between each sensor. Correlation of sensor 1 with sensors 3 through 5 have strong coherence at 0.7 Hz and 1.25 Hz (almost the modulated plume coherence). Correlation with 1 and 7 through 9 have a peak at 0.85 Hz and all correlations peak at 1.7 Hz.

In Fig. 5.19, a comparison of the array size and the correlation between the sensors is examined. It is interesting to note that between 0.5 cm and 2 cm, the coherence linearly decreases but the frequency peaks generally remain the same at the lower frequencies. Over 2 cm, the peaks slightly shift and the linear decrease is only seen at low frequencies.

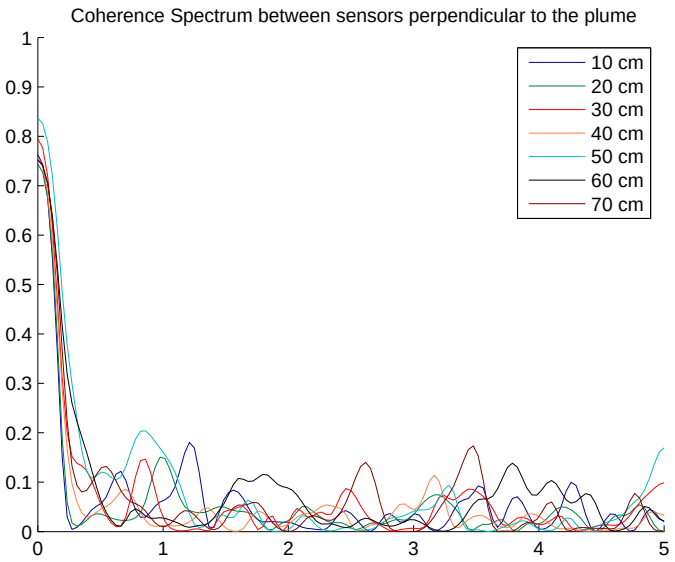


Figure 5.17. Comparison of coherence of sensors 1 and 4 of an array at (205,100:100:700), long range distances.

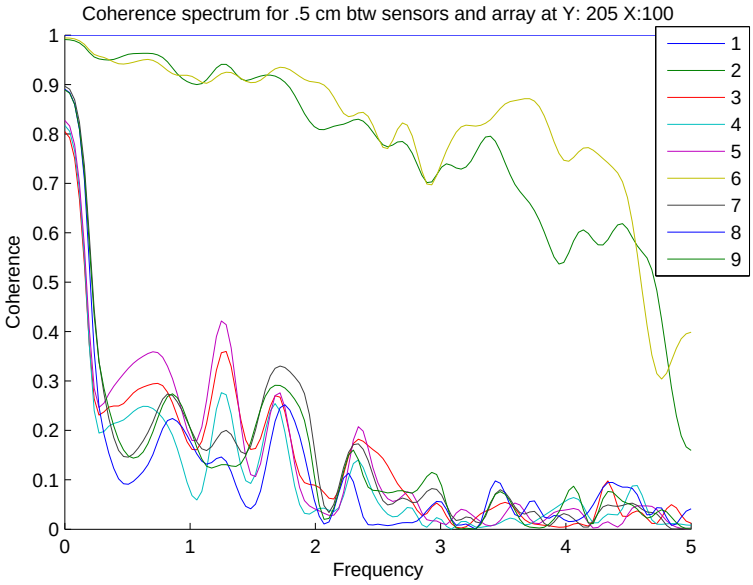


Figure 5.18. Example of 0.5 cm between each sensor in the array (205,100).

5.3 Exploiting Phase - Correlation Analysis

Up until this point, the magnitude of the FFT, the power spectral density, and the magnitude of the complex coherence of the signal has been analyzed. There is a key measure, of utmost importance in spatial array signal processing, missing in these analyses, namely the

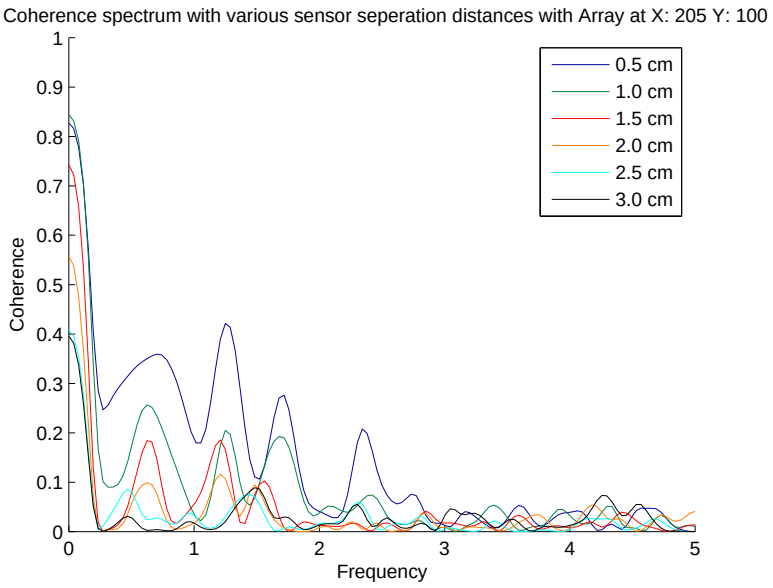


Figure 5.19. Comparison of various array sensor distances when the array is at the same center location (205,100). Coherence between sensors 1 and 4 are shown.

time-delay (or phase) information. With a simple time delay measure, we can determine the speed and direction of a propagating wave (e.g. acoustic velocity). See Appendix A for more information.

5.3.1 Correlation analysis

To obtain time delay information, we will look at the correlation lags between sensors which are obtained from the peak delays between the autocorrelation function (5.12) and the cross-correlation function (5.4). For initial evaluation, we take cross-correlations of the entire time window of the data which is 600 seconds. In this section, an angle-of-arrival (AOA) estimate is derived from all 600 seconds, while in Section 5.5, the window correlation length is shortened to show feasibility for a real-time implementation.

We first placed the array at (200,120), then took correlations with all the sensors (the setup can be seen in Fig. 5.13). When zoomed out, the correlation looks like Fig. 5.3.1. As expected, the signal waveforms are most correlated when lined up with each other in time and then slope off correspondingly. Let's now take a closer look at the middle. When we zoom in, as in Fig. 5.21, we see that there are peaks at various correlation lags. The

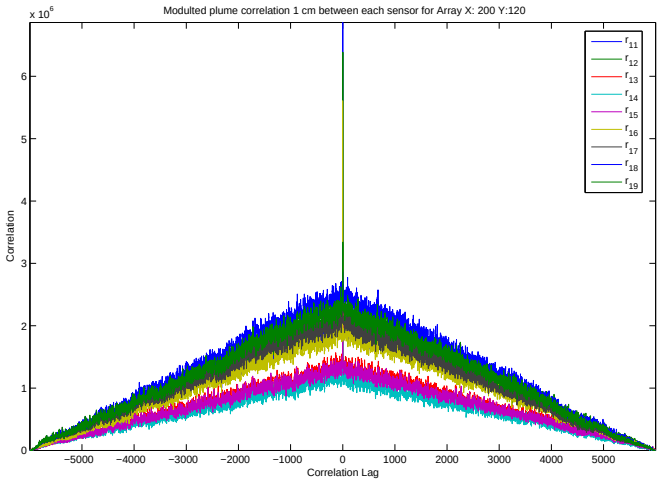


Figure 5.20. Correlations of the center sensor with the 8 sensors around it, over 600 seconds.

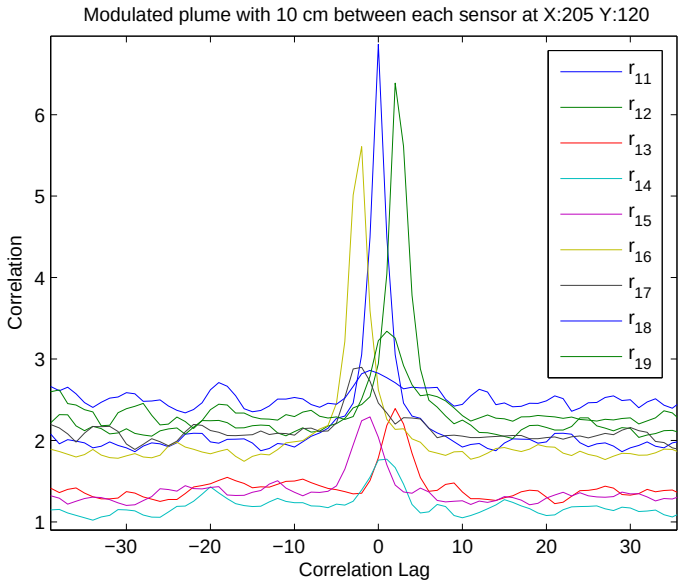


Figure 5.21. Correlations of the center sensor with the 8 sensors around it, over 600 seconds, zoomed in to see the lowest correlation lags.

correlations lags tell us the time difference of the lined up signals that have the highest correlation. From this, we can determine that the signal reached sensor 6 two timelags before it reached sensor 1.

We will illustrate how such information can help determine the source of the plume. If we use the time delay information to weight each sensor location and then average the

weighted sensor coordinates, we can obtain the direction estimate via:

$$(x, y) = [\mathbf{xy}]^T * \mathbf{delays} \tag{5.11}$$

where **x** and **y** are the x and y coordinates of the sensor array and **delays** are the time lags of each r_{1x} peak.

Using this, we illustrate our localization using (5.11) for a sensor array placed at position (200,120) using 1 cm between each sensor in the modulated and unmodulated plumes. Later, we will expand and show the localization result and using 1.5 cm between each sensor. We then repeat these scenarious for the (205,400) location. These results can be seen in Fig. 5.22. In (a) and (d), the sensor separation is 1 cm and it is tested on an unmodulated plume. In (b) and (e), the sensor separation is 1 cm and tested on the modulated plume. In (c) and (f), the sensor separation is 1.5 cm and placed in a modulated plume.

What we are able to see in Figs. 5.22 (a), (b), and (c) is that this measure does not work well for the unmodulated plume but works a little better for the modulated plume, but loses its ability to track when the array is larger. This is mainly due to the fact that the correlations between sensors are not that high. We did find that the closer the sensors are together, as seen in (b) as opposed to (c), the better the correlation. Also, the time correlations did not yield peaks as high peaks for the unmodulated plume when we compare the modulated (a) vs. (b). All in all, the sensor array placed 28 cm further down the plume field was able to compute a better estimate than the one closer, and was robust in all the scenarios we tested. This may be due to the fact that the concentration was not as high as before and that the few characteristics that exist in the field at this farther point exhibit prominent correlations.

For comparison, we will plot the corresponding correlation vs. lag plots for each of these scenarios in Fig. 5.23. As one can see, when clear peaks are not seen in the correlations, and a true time-delay cannot be determined, a stray peak is chosen (sometimes at a very long lag such as (c)), and this throws off the crude estimate. The advantage of enlarging the array is that instead of getting correlation lags that resolve to 1 or 2 timesteps as in (e), we obtain better resolution and get a range of 1-6 timesteps in (f). However, the

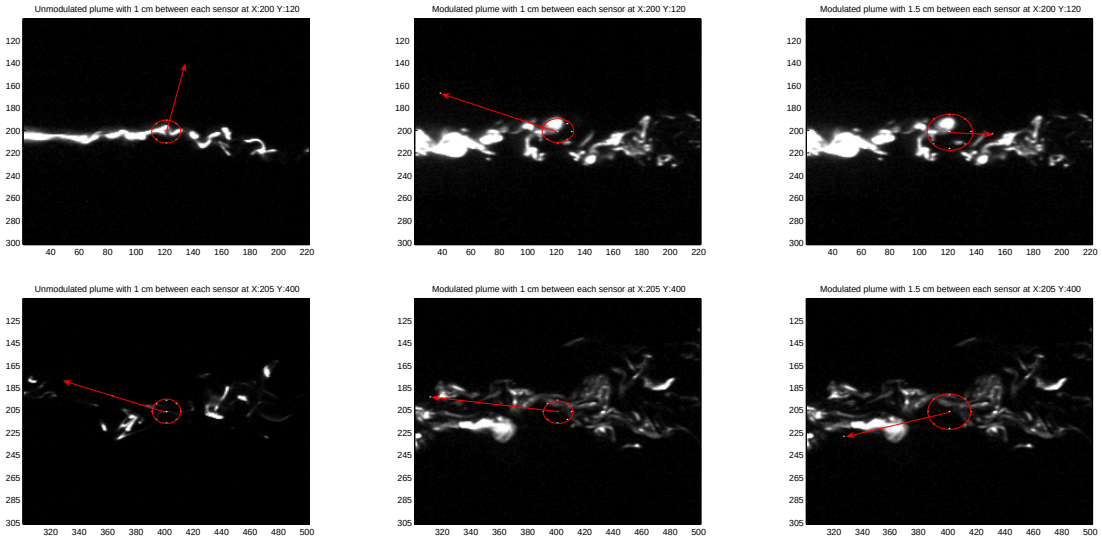


Figure 5.22. a-c (top row),d-f (bottom row): The top row is the sensor array placed at (200,120) and the bottom sensor array is placed at (205,400).

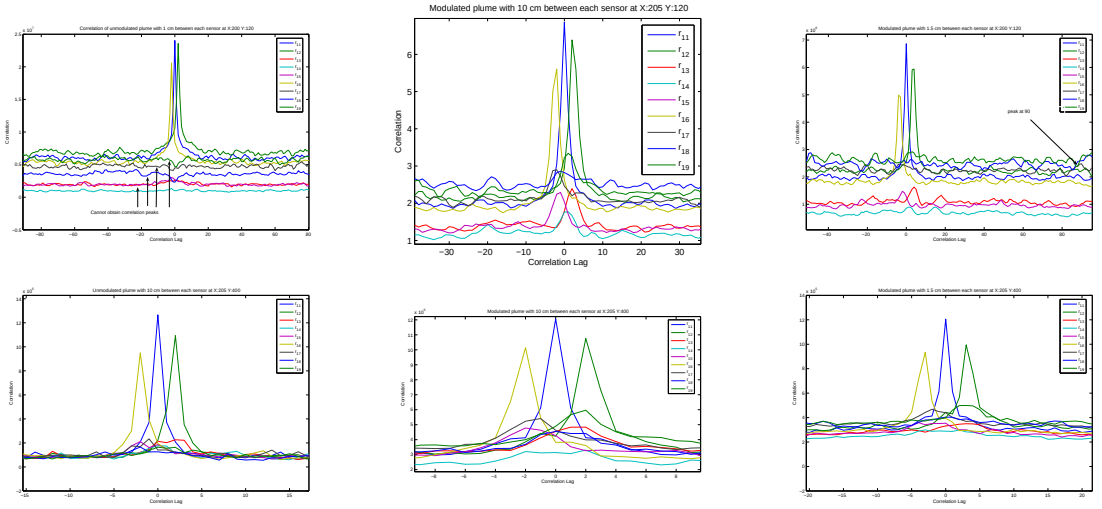


Figure 5.23. a-c (top row),d-f (bottom row): Corresponding time correlations to Fig. 5.22

trade-off is that the peaks are not as clear.

As we have expressed before, what we have illustrated in this section is the time delay information and introduced a crude computation to exploit it. More advanced techniques can be developed – for example, we would probably want to compute the short-time correlations on short windows rather than the whole 600-second sequence and then average all direction-of-arrival estimates. This is explored in Section 5.5.

5.3.2 Matching the measurements to the data

The units of the experimental data are as follows. The experimental mean flow velocity was 50 mm/s. The frame capture rate of the camera is 10 Hz. Each pixel in the modulated plume represents 1.0345 mm, and each pixel in the unmodulated represents 1.0417 mm.

So, what is a correlation lag? Since a correlation lag just means the two segments are separated by one time step, in this case 0.1 seconds (inverse of the frequency rate). To recompute our measurements for Fig. 5.21 to see if it is detecting the correct flow of the field, we can say that the time delay of the sensor 6 to sensor 1 is 2 time lags or 0.2 seconds. The sensors are 10 pixels apart so they are 1.0345 mm*10 = 10.345 mm apart. So the plume covered 10.345 mm in 0.2 seconds yielding a 52 mm/s flow rate which is approximately the 50 mm/s reported in the experiment.

The choice of the sensor separation and the frame rate should be taken according to what flow rate detection is needed. This means that if a "leak" is too fast, closely spaced sensors will have too high of correlations and less resolution. Also, if the flow is too slow and turbulence is the dominating factor, sensors placed too far apart will yield low correlations as well. So, while correlations can tell much about data, they must be designed for a specific flow.

Before performing our sensor localization measurement, we also analyzed the effect of how far in the plume the sensor measurements were correlated. Placed around (200,500) in the plume, we found that sensors placed linearly parallel to the plume (see Fig. 5.25) showed correlation peaks for up to 120-140 cm between them. When 1 cm is between each sensor, the 9th sensor peak is barely intelligible. When 1.5 cm is between each sensor, the 9th sensor is pretty much in the noise level. Finally for the 2 cm separation between each sensor, the 8th and 9th sensors are in the noise. So, rather than the circular array used in this section to determine the direction of the plume, a star array (or a several linear arrays) would be more effective in gaining phase information from the plume. And now that we know the range of the phase correlations, we could intelligently place the sensors.

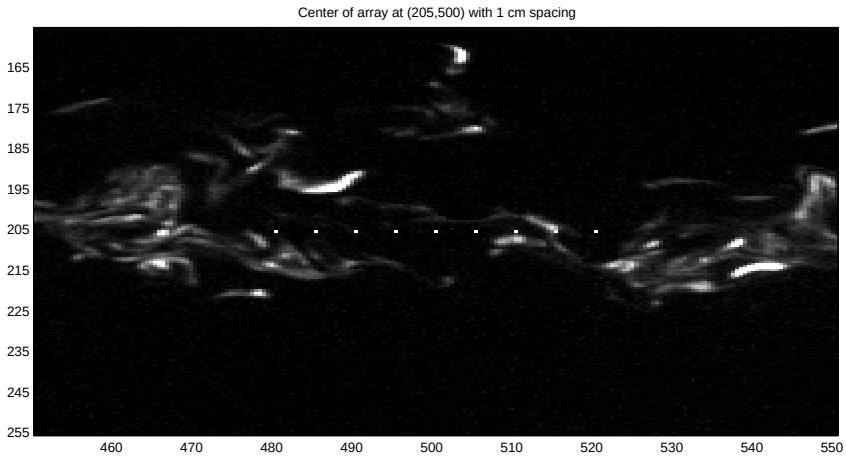


Figure 5.24. Linear array on the centerline of the plume, 50 cm away from the source.

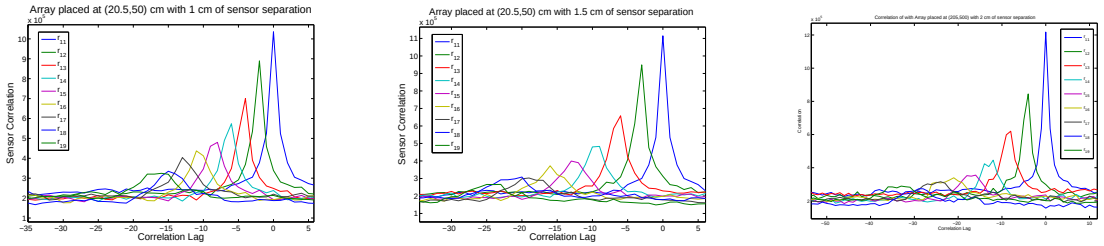


Figure 5.25. a)-c): Time correlations of a linear array with 1 cm, 1.5 cm, and 2 cm between each sensor.

5.4 Short-time Analyis and Modelling the Spatial Linear Filter

In this section, we discuss an interesting result that came about from our quest to find a unique 1 Hz frequency as found in [45]. As seen in section 5.2.2, if input/output signals have the linear relation, (5.7), then result of minimizing the Wiener-Hopf equation gives us a short-time filter in the frequency domain as (5.8). But these equations are only valid, if there is no mean squared error (MSE). In (5.10), we showed that if the magnitude of the coherence is 1, there is no MSE.

From our data analysis, we found that if we take a short enough window (usually 6 seconds or less), the magnitude of the coherence is one between sensor 1 and each sensor. Thus, we can then model a spatial filter between each sensor. One might ask why this would be interesting. Firstly, if the plume can be modelled as a linearly time-varying system, this opens up a wide range of analysis, (e.g. known to speech processing). Secondly, if we can

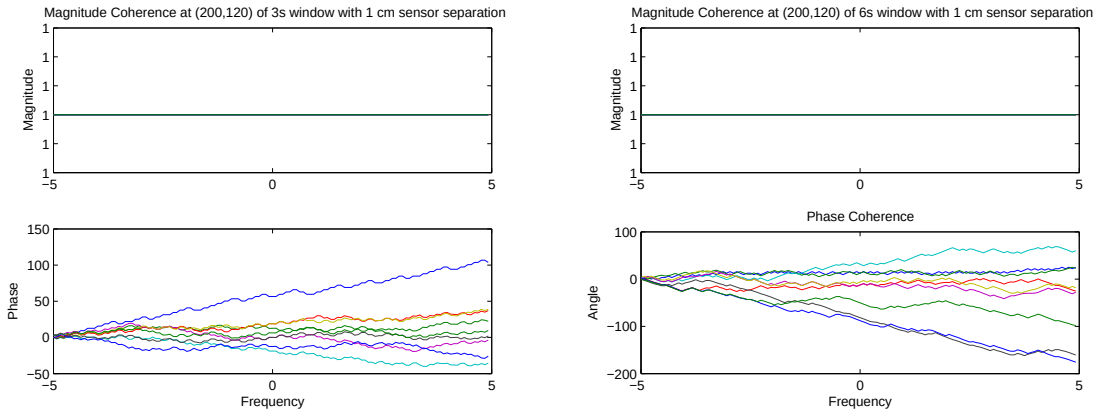


Figure 5.26. a): Short-time coherence, 3 second window, of the sensor array placed at (200,120) with 1 cm sensor separation. b) the same except with a 6 second window.

model the spatial filter between sensors, we can then start to view how the plume changes over time (e.g. characterize it on a short-time basis then observe slow changes).

Again, we use the sensor array found in Fig. 5.13. In Fig. 5.26, the coherence using two different windows, one being a 3 second (or 50 pixel) window and the other being a 6 second (or 100 pixel window), is shown. From these complex coherence plots, we can see that the magnitude coherence is 1 and it has **fairly** linear phase. Using (5.10), we can now say that (5.8) holds true. If we take small windows, we can model the plume dynamics as linear time-varying system. The fft length is 256 to get the coherence plots.

Equation 5.8 is then used to determine this filter for the window. The plots corresponding to the coherence plots are seen in Fig. 5.27. They illustrate not only one short-time window in a) and b), but all of the windows averaged as seen in c) and d). For the 3 second window, 200 windows are averaged while for the 6 second window, 100 windows are averaged.

Fig. 5.28 illustrates how placing the array at various locations (200,120), (205,400), (205,800) changes the filter. It almost looks as though the filter peaks slightly shift in frequency as the location changes. This should be investigated more, but due to the diffusion aspect, the frequency of the karman vortices may elongate with distance from the source.

Finally, we examine the effect of varying the distance between sensors on the resulting

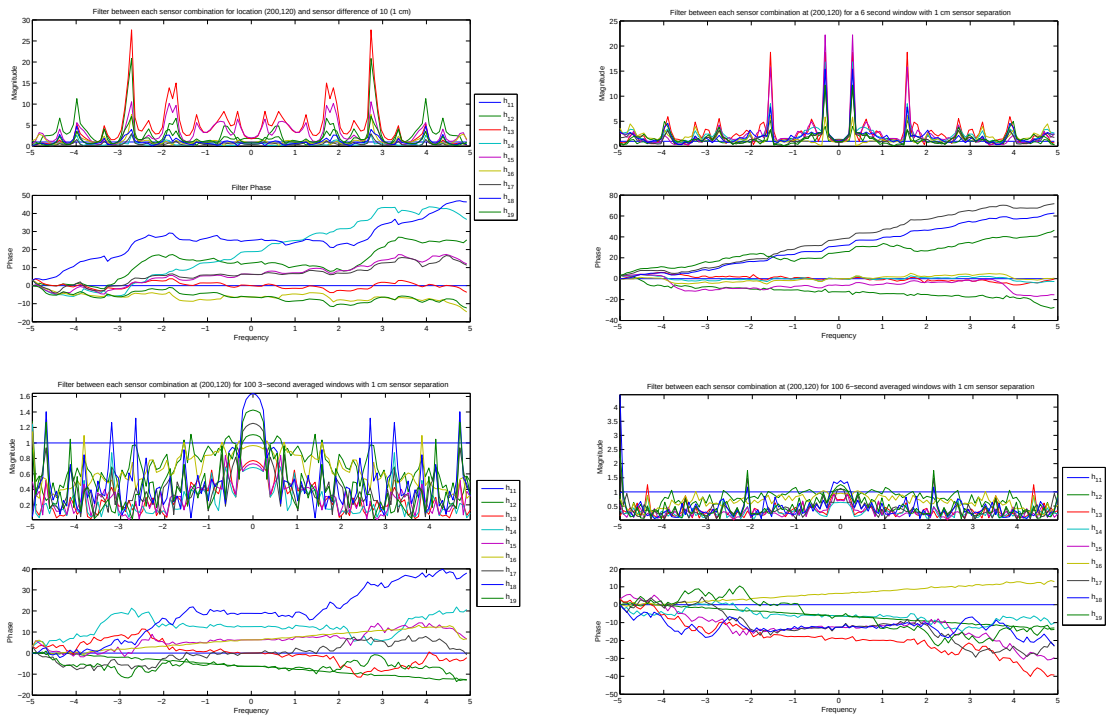


Figure 5.27. For the sensor array placed at (200,120) with 1 cm between each sensor, the top row, a) and b), are the filters for one 3s and 6s window. In c) and d), they are the averaged windows over 600 seconds for the two window types and same setup.

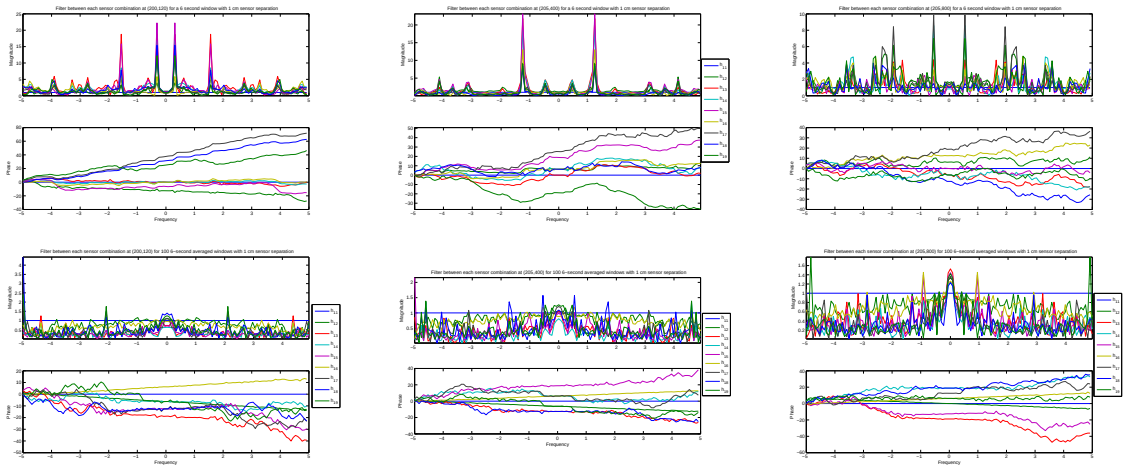


Figure 5.28. a-c (top row),d-f (bottom row): With 1 cm between each sensor in the array and 6 second windows, the top row illustrates the filter variation due to one window as the array is placed further away from the source. The bottom row shows the distance effect on the filter from the average of the windows.

"averaged" filter in Fig. 5.29. The averaging in this section means that the filter for each window is summed and then divided by the number of windows. Nothing conclusive can

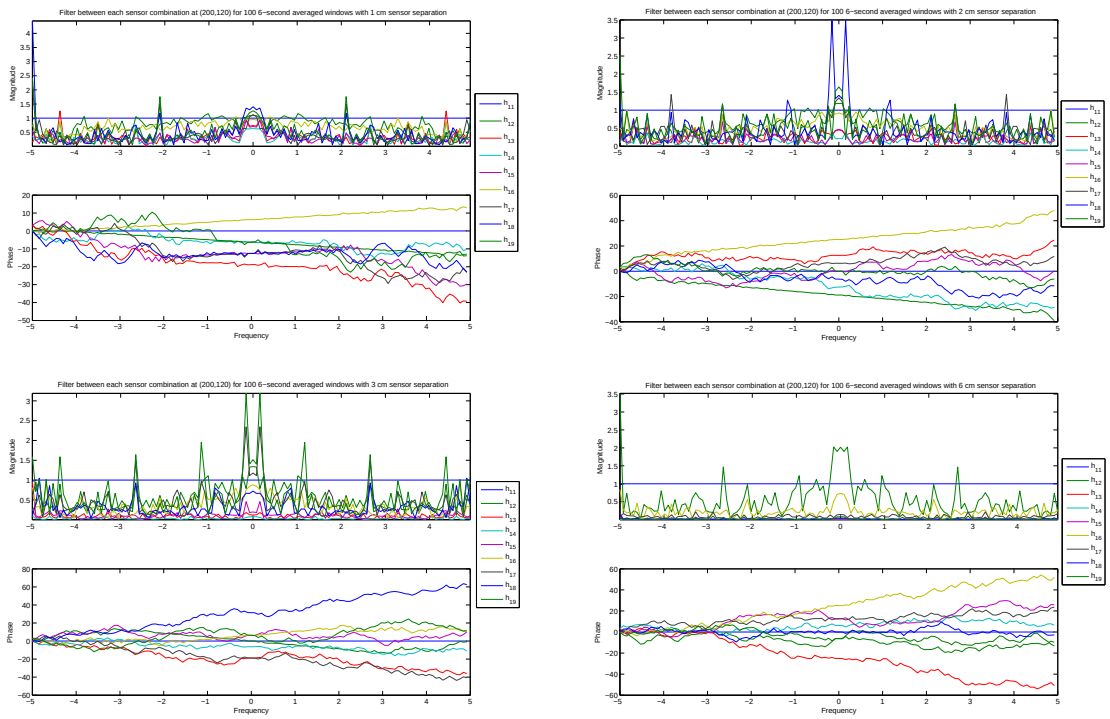


Figure 5.29. For all of these graphs, the sensor array is placed at (200,120) and for the averaging, a 6s window is used. a) is when a 1 cm, b) 2 cm, c) 3cm, d) 6cm spacing separates the sensors

come from here except that it is interesting that once we go to 2 or 3 cm, a peak arises at the low frequencies compared to the 1 cm case. The 3 cm seperation seems to yield some interesting harmonics, but going to 6 cm yields nothing except there is much more difference in the energy of the various sensor pair filters than before due to diffusion; the filter for sensor 1 to sensor 2 has the best energy. Although the results from these graphs are not obvious, we believe there is an optimum sensor separation (depending on the leak rate) that will model the plume the best.

5.5 Chemical source localization in unknown turbulence using the cross-correlation method

Estimating the direction of a diffusive source is a difficult problem, and little has been tried to estimate a chemical source subject to turbulence. Turbulence must be addressed if chemical localizer systems are to be effective. We look at how to quantify turbulence and develop a measure to locate a source in two different types of turbulence, modulated and

unmodulated plumes. We show that a plume can be modeled linearly on a small-scale and that a wind measure for a stationary sensor array can indicate the direction of a chemical source with reasonable accuracy and time. This measure can be easily implemented in low-power computational electronics and applied to the detection of chemical leaks and illegal substances.

Recently, engineers have begun addressing the hard task of diffusive source localization ([70], [84], [103]), but diffusive fields are rarely found in natural atmospheric conditions. Usually, turbulence is also a factor which further compounds the difficulty of the chemical localization problem. This is due to the fact that turbulent advection disperses a chemical and causes discontinuities in the flow. To attack this problem, most approaches use mobile sensing robots which can survey and sample a large plume. In [28], an intelligent plume mapping scheme based on HMM's for an autonomous vehicle is devised. In [46], a transient-response-based algorithm is used to in several modes, one being to track the plume upwind and another to switch into a local search.

Only mobile and multifaceted algorithms have been implemented because they are able to perform well for tracking sources in nonlinear, dynamical plumes. However, these solutions are complex and difficult to implement. In this chapter, we show that an easy-to-implement stationary array can localize the direction of a source in two different turbulent scenarios in a matter of minutes.

Coherence spectra has been used to detect distance from a chemical source in plumes [50], but this measure loses all spectral phase information about the plume needed for source localization. In this chapter, we will determine the flow of the plume by exploiting the sensor time delays. If a source releases a chemical at a constant rate and if we use an array small enough compared to the turbulent parcels (pockets of high concentration) in the plume, a rough estimate of the source location can be determined from computing the angle of arrival (AOA) of the wind direction.

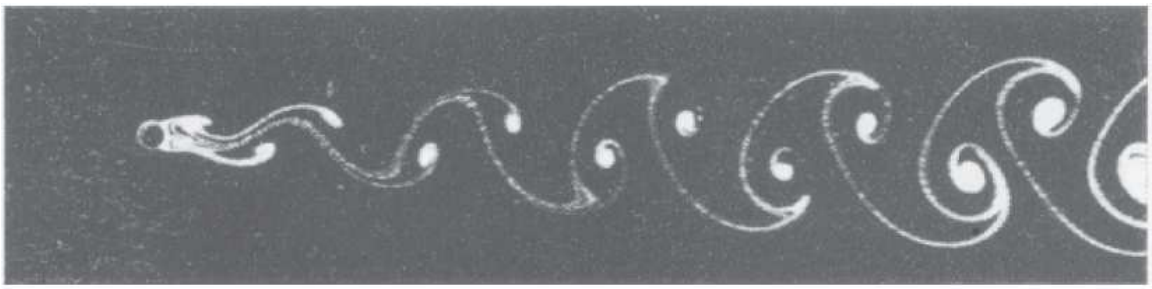


Figure 5.30. An ideal von Kàrmàn vortex street with Reynolds number, $R = 73$. [11]

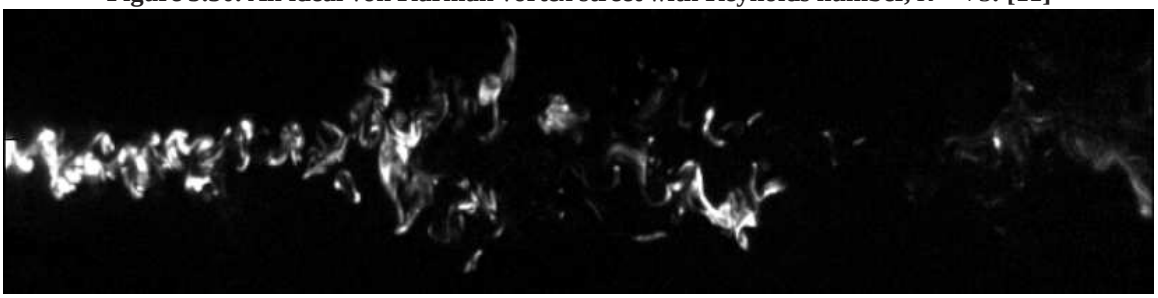


Figure 5.31. Our modulated plume (von Kàrmàn vortex street) data with the Reynolds number above 1000. The modulated turbulence dissipates due to the effects of natural turbulence and diffusion.

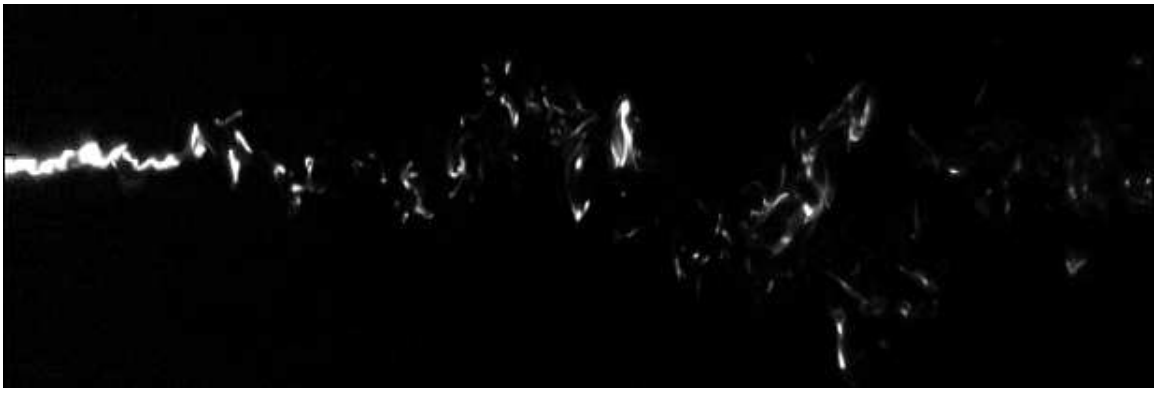


Figure 5.32. Our unmodulated plume data; the transitition from laminar to turbulent flow occurs due to natural turbulence and diffusion.

5.5.1 Assessing the turbulent data

A simple model of diffusion can be written as (4.4). We do not only have effects from diffusion in our problem, but we also have unmodulated turbulence from a pure chemical flow at 5 cm/s in one case (Fig. 5.32). The unmodulated plume is subject to basic diffusion and turbulent advection. In the second case, we have a von Kàrmàn vortex street, or modulated

turbulence, from a 5 cm/s chemical flowing over a 0.8 cm diameter cylinder (Fig. 5.31). This modulation compounds the normal turbulence to create more unpredictable effects. Our data was collected with planar laser-induced fluorescence (PLIF) technology which is a non-intrusive, optical measurement technique to obtain a sequence of instantaneous, high-resolution spatial concentration fields. The plume is sampled at a 10 Hz framerate, each pixel represents approximately 1mm $\times$ 1mm in space, and the concentration values are normalized and quantized to values between 0 and 255.

When a chemical flow passes around a cylinder, a von Kàrmàn vortex street results, and an ideal one is seen in Fig. 5.30. Figs. 5.31 and 5.32 are single frames captured from the turbulent plume data. In Fig. 5.31, there is slight von Kàrmàn vortex shedding in the modulated plume, but it is subject to many diffusive and additional unknown turbulent forces. In Fig. 5.32, the unmodulated plume is subjected solely to diffusion and natural turbulence. These images are a cropped version of the full 401×940 pixel (y-dimension $\times$ x-dimension) images. The centerline of the plume is 205 pixels (20.5cm) down on the image.

While we have a model for the diffusion, we do not know the exact models for the turbulence or the Reynolds number of the vortex shedding. This makes attacking the problem challenging because we are blind to the turbulence involved. Therefore, we desire to localize the source of the plume using *only* the following knowledge:

- Intermittent "events" occurring in the plume due to turbulence
- Constant flow rate
- Diffusion dissipating the events after a certain distance

5.5.2 Cross-Correlation method for Wind AOA

Suppose we have N sensors and a source signal, $s(t)$ propagating through air. Due to Fick's second law, an unknown nonlinear turbulence function, $f(\ )$, and sensor noise, the signal

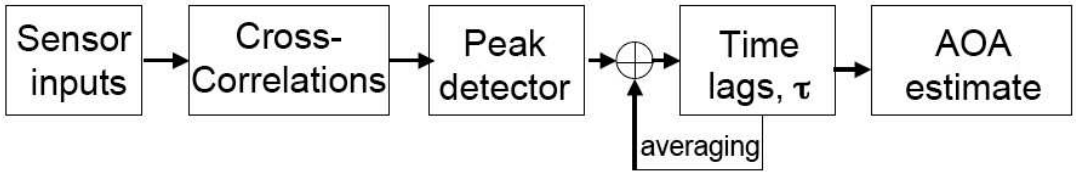


Figure 5.33. Block diagram of the proposed wind AOA algorithm.

received by the i^th sensor can modeled as:

$$x_i(t) = f(\frac{s(t)}{r_i}) + n_i(t)$$

where r_i is the distance from the source to the ith sensor and $n_i(t)$ is additive white Gaussian noise.

In (5.12), we test to see if linear correlations exist between sensors in a window of size T, assuming that the $f(\,)$ signal and n are not correlated.

$$r_{1i}(t) = \sum_{k=0}^{T} x_1(t)x_i(t+k) \tag{5.12}$$

Then the delay that maximizes the peak of the function is stored in τ_i. Column vector, τ, contains all the peak delays between the N sensors.

If the array has a 2-D symmetrical geometry, τ can be used to estimate the wind direction by weighting the coordinates with the delays. We now represent the array center with cartesian coordinates $\mathbf{x} = [x, y]^T$, and the sensor coordinates in reference to the center are:

$$\mathbf{X} = \begin{bmatrix} x_1 - x & x_2 - x & \dots & x_N - x \\ y_1 - y & y_2 - y & \dots & y_N - y \end{bmatrix}.$$

Next, the direction of the wind from the centroid of the array is estimated as:

$$\begin{aligned} [d_x\, d_y]^T &= \mathbf{X} \cdot \tau \\ \theta_{wind} = \text{atan}(\frac{dy}{dx}) \quad \theta_{source} &= \text{atan}(\frac{-dy}{-dx}) \end{aligned} \tag{5.13}$$

We denote θ_{source} as the angle of arrival (AOA) and $\tau[n]$ as the sensor delays for each window. $\tau[n]$ is averaged over time by $\hat{\tau} = \frac{1}{N}\sum_{n=0}^{N-1}\tau[n]$ to smooth the delay estimates

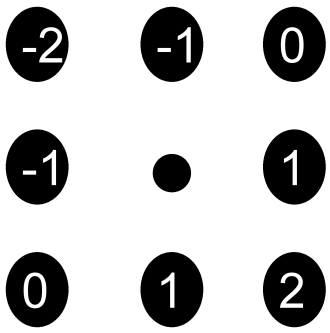


Figure 5.34. Illustration of a 135° source localization scenario for an $N = 8$ sensor array. The numbers on the sensors are the τ_i's corresponding to the time delay with respect to the center sensor. Weighting the coordinates with these values, gives us a wind localization of -45°. The opposite direction is taken as the source direction.

where L is the entire data length, and $N = L/T$. Then, using (5.13), a time-averaged $\hat{\theta}_{source}$ is determined.

A block diagram of the algorithm can be seen in Fig. 5.33.

5.5.3 Numerical evaluation for 2-D stationary arrays

Now, we want to expand our sensor array from a linear to a symmetric 2-D array in order to estimate the direction of a source. Using a linear array for this task is difficult since this is not a wavefield; only intensity information is available. We design a square array of $N + 1$ sensors where N is the number of sensors that maximize the perimeter of the array, and one sensor is placed at the centroid of the array. In (5.12), we assign this middle sensor as the first sensor, and compute the correlation of it with the surrounding i sensors to obtain τ. τ can help us determine the AOA (see Fig. 5.34).

With our algorithm, there are a few parameters to take into consideration with each plume:

- The distance between sensors / array size
- The sensor array placement in the plume

First, we examine the array size, or the distances between sensors, to localize a source; see Fig. 5.35 to get an idea of the scale of a "large" array that we used in the plume. We

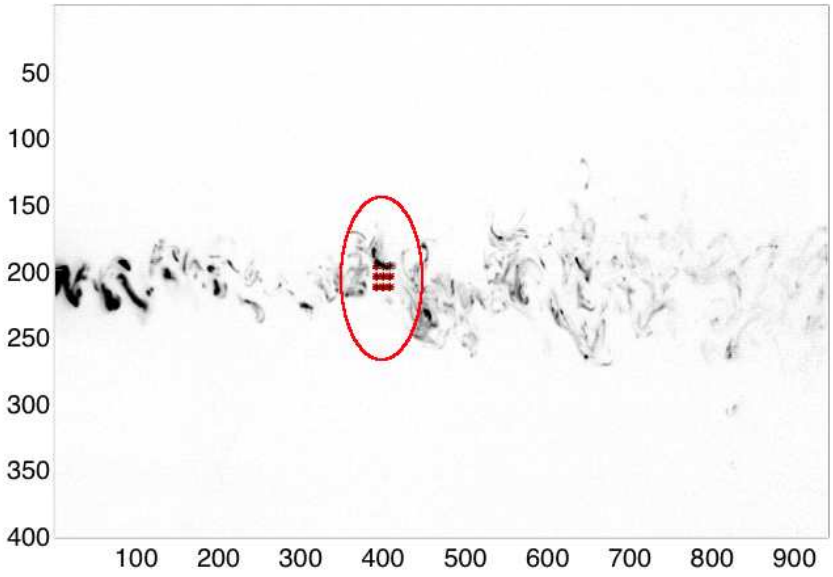


Figure 5.35. An $N = 8$, 1.6cm $\times$ 1.6cm array placed at $(20.5, 40)$cm in the modulated plume.

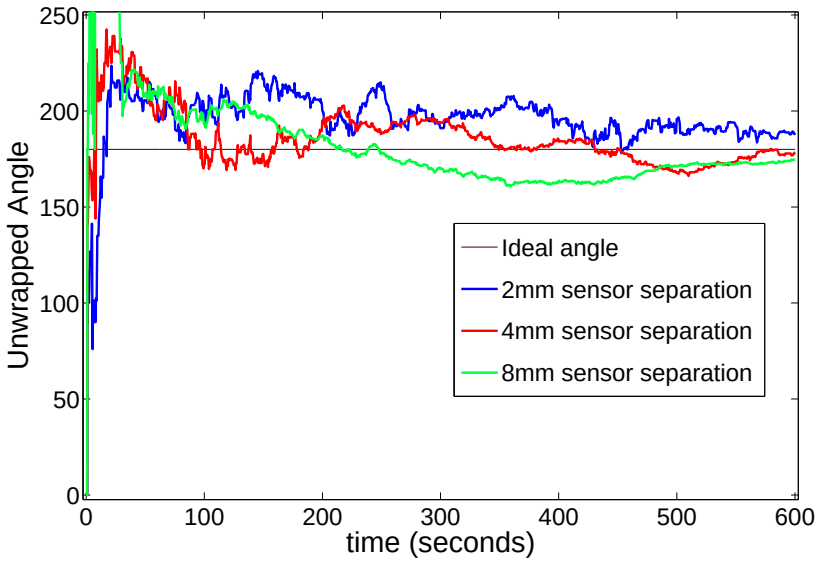
also want to examine the difference between localizing the unmodulated vs. modulated turbulence. Visually, the plumes do not look considerably different, but the AOA estimate converges much slower for the modulated plume. The comparison of the modulated vs. unmodulated effects can be seen in Fig. 5.36.

Ideally, we want to be able to blindly locate a turbulent source despite the sensor array orientation in the plume. Two strenuous array placements are tested. One tests the array's ability to track the plume while being placed a bit outside the flow (see Fig. 5.37). The second tests the ability of the array to localize the wind direction while being far downstream from the source (see Fig. 5.38).

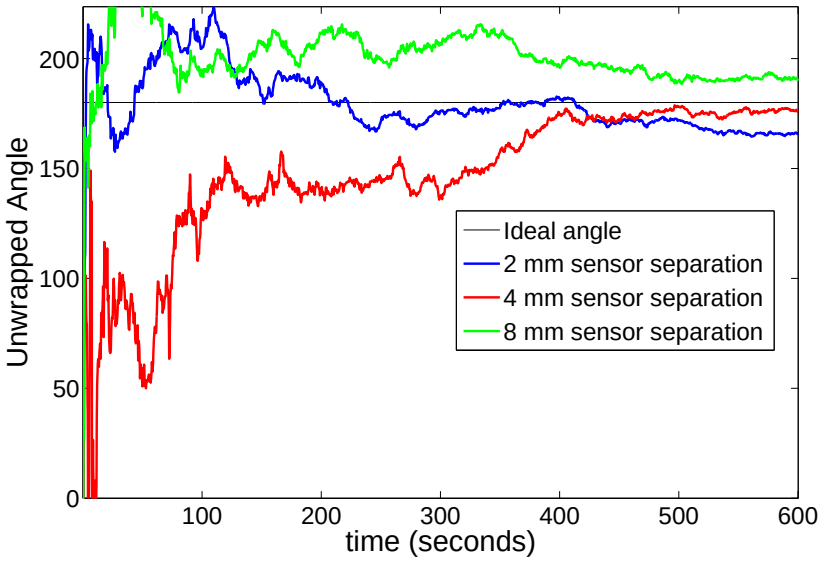
Finally, distance and plume-type comparisons are shown in Table 5.1.

5.5.4 Conclusions

This chapter demonstrates that nonlinear turbulent plumes can be linearized on a small-scale and that we can exploit the time delays between sensors with the cross-correlation method to obtain wind direction and chemical source AOA estimates from the plume. With



(a) Unmodulated plume.



(b) Modulated plume.

Figure 5.36. The effect of the array size/lateral sensor separation on the convergence time of the source AOA. The array's center was placed at $(20.5, 40)$**cm (**180° **angle from the source), the window correlation length is** 0.5 **seconds, and the array has** $N = 8$ **(**8 **sensors on the perimeter and one in the middle). Clearly, the algorithm converges slower in the (b) modulated plume compared to the (a) unmodulated plume.**

a stationary array placed 80cm away from the source in modulated turbulence, the algorithm converges to within 90% of the AOA in approximately 400 seconds (and to 80% of the AOA in about 200 seconds). Localization time in unmodulated turbulence takes about

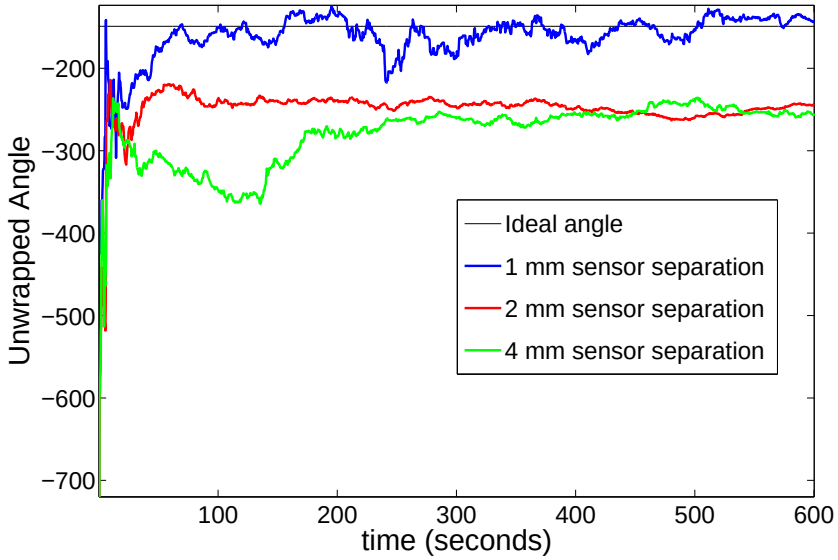


Figure 5.37. The sensor array placed at $(17.5, 5)$cm **in the modulated plume, with the source** $-150°$ **from the array.** $N = 16$ **in this case, and the correlation window length is 0.5s. In this case, the larger the sensor array, the worse the performance of the localization. This is due to the fact that pockets of concentration are closely spaced when near the source, and if the sensors are too far apart, they are uncorrelated.**

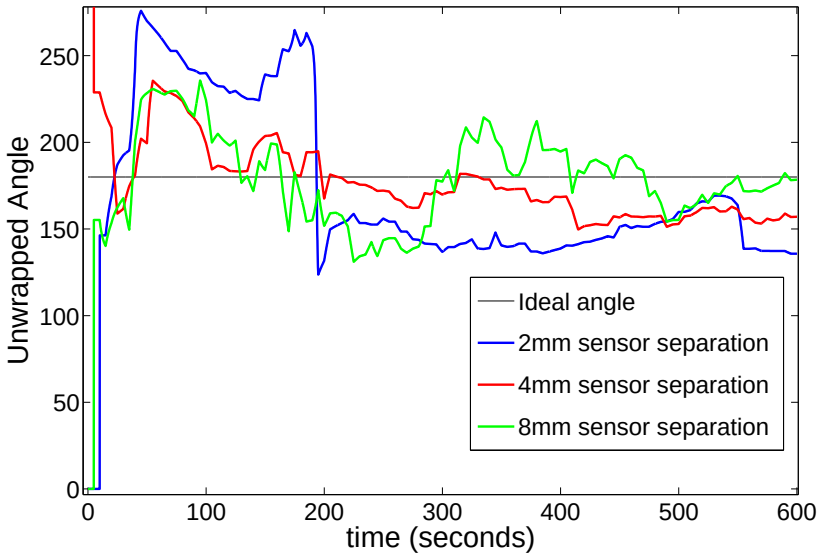


Figure 5.38. The sensor array placed at $(20.5, 80)$**cm, a** $180°$ **angle from the source, in the modulated plume.** $N = 8$ **in this case, and the correlation window length is** 0.5**s. In this case, enlarging the array size improves the array's localization time and estimate of the source angle, due to the fact that the pockets of concentration are dispersed and greater in size.**

half the time of the modulated convergence time on average. In the future, more controlled

Table 5.1. A table summarizing the average final angle error and the convergence time to reach 90% of the final angle. Here, $N = 8$, the correlation window length is 0.5 seconds, and the results from the 2mm, 4mm, and 8mm sensor separations are averaged for each placement/plume. The convergence time doubles in a modulated plume over an unmodulated plume. Placing the array twice the distance from the source, the convergence time again almost doubles. The angle error also has a linear increase with these scenarios.

Position	Unmodulated/ Modulated	Avg. Final Angle Error (in °)	Avg. Time to 90% of Final Angle
(20.5,40) cm	Unmod	6°	120s
(20.5,40) cm	Mod	10°	225s
(20.5,80) cm	Mod	13°	420s

turbulent scenarios should be explored. For now, this stationary solution is a simple measure compared to mobile implementations and can be easily implemented in low-power computational electronics for many security applications.

5.6 Localizing direction-of-arrival in unknown turbulence using delay-and-sum beamforming

In the previous section, we show that an AOA can be obtained from pairwise sensors in an unstable plume, as long as the sensors are relatively close (e.g. $9 - 10cm$ for this plume). There are other techniques to exploit time delay of arrival (TDOA) information. One such technique we explore in this section is delay-and-sum beamforming using a uniform linear array. As shown in Appendix A, if a planar wave passes over a linear array, the time delay between sensors is related to the AOA, via $\tau = \frac{d}{v} sin(\theta)$ where d is the distance between sensors, v is the constant wind velocity, and θ is the AOA.

In beamforming, these delays can be exploited to get better gain of the signal by lining up the waveforms in time. Conversely, the TDOA of an incoming signal can be obtained from the delay with the largest gain. A broadband delay-and-sum beamformer is illustrated in Fig. 5.39. In this case, when $T = \tau$, the channels are all time aligned for a signal from direction θ, and when time aligned, $y(t)$ will have maximum gain. So if the value of τ is unknown, various values can be tested and the τ that yields the greatest $y(t)$ gain will be the τ that time aligns the signals. Such methodology is followed in this section, and the τ, or θ

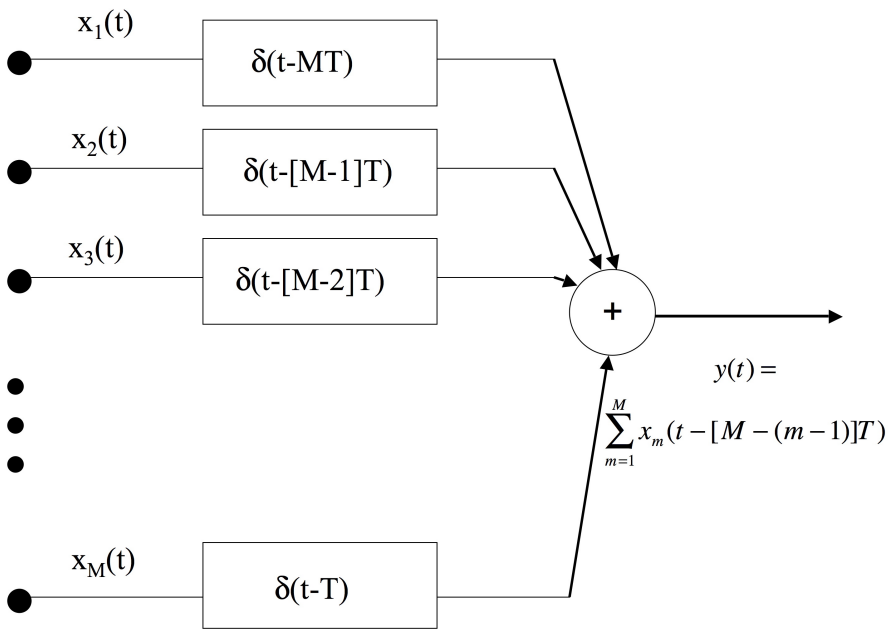


Figure 5.39. A vertical uniform linear array with M input sensors and gain $y(t)$.

(AOA), to maximize the gain is sought from plume sensor measurements.

Each pixel in the dataset represents approximately 1 mm. The modulated plume is the data set used in this section, which is sampled at a rate of 10 Hz and is flowing at approximately 5 cm/s (5 pixels per time sample).

We test one such linear array in a horizontal arrangment shown in Fig. 5.40. First, the delay-and-sum beamformer is tested over several τ, in this case each τ is the time length of one sample, 0.1 s, shown in Fig. 5.41. We interpolated the time axis so that each time sample shown now corresponds to 1/100 of a second but still tested over integer τ; the interpolated graph is shown in Fig. 5.42. From these figures, it is shown that a clear gain exists between τ's of 0.4 and 0.5 seconds continuously through time. There is 20 mm between each sensor, and the plume is travelling at approximately 50 mm/s, experimentally reported in the Hiroshi's dataset [45]. Taking sensor spacing and dividing it by the wind velocity, it should take approximately 0.4 s to pass from one sensor to the next. The Gain vs. Delay graphs may indicate that the wind velocity or direction-of-arrival might be fluctuating between 40 and 50 mm/s, and since the data taken was the turbulent

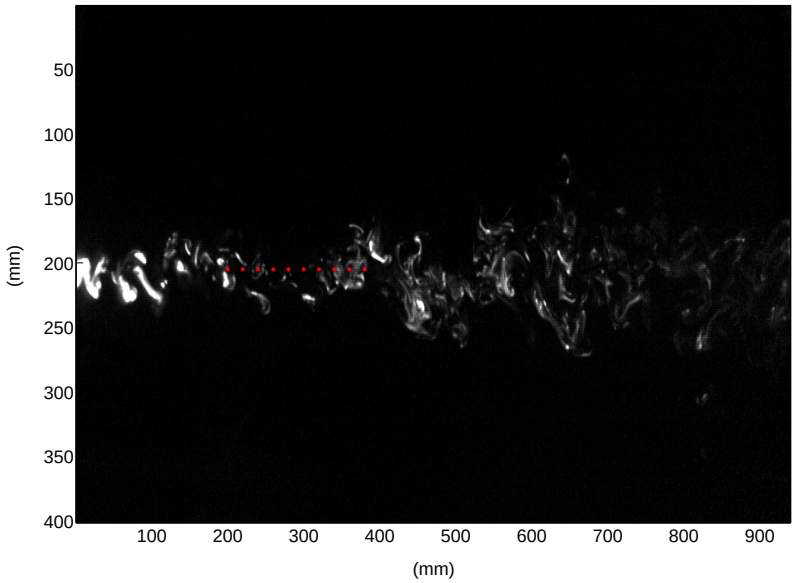


Figure 5.40. A horizontal linear array with 10 sensors, with 2 cm (20 pixel) spacing, starting at 20 cm down the plume on the centerline.

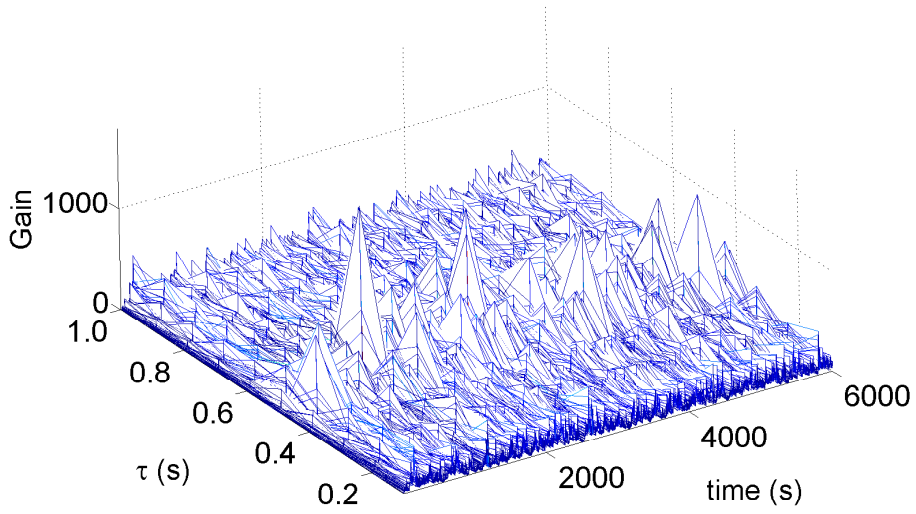


Figure 5.41. A plot of the gain, $y(t)$, vs the delay (in seconds) vs. time in seconds for the sensor arrangement seen in Fig. 5.40. The highest gains are between 0.4 and 0.5 seconds, which correspond to approximately the delay caused by the 2 cm sensor spacing.

plume, this is likely to be the case. Taking the maximum of the gain values over all time in Fig. 5.43, we can see that the highest gains lie at 0.4 and 0.5 seconds. Taking a fine sampling of τ, seen in Fig. 5.44, we can see that the maximum τ values can lie anywhere

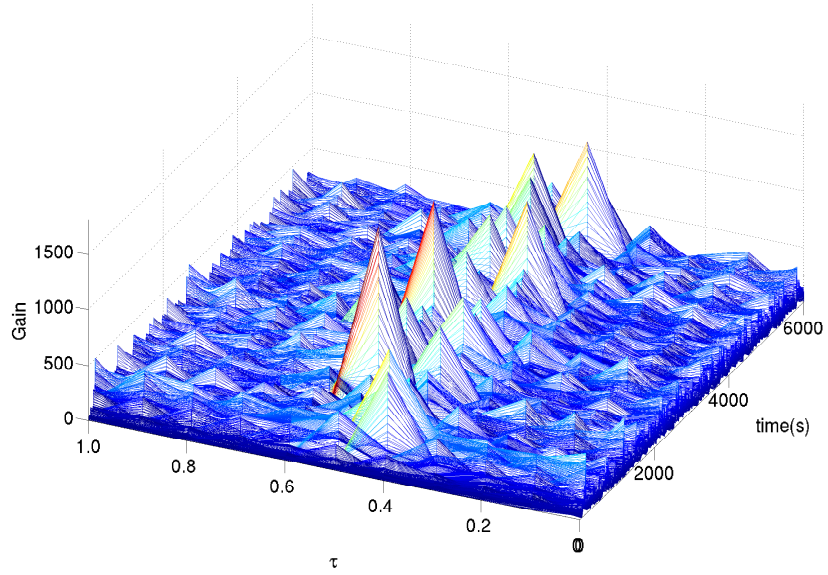


Figure 5.42. A plot of the gain, $y(t)$, vs the delay vs. time with an interpolation of the time data by ten for the sensor arrangement seen in Fig. 5.40. Integer τ are still tested.

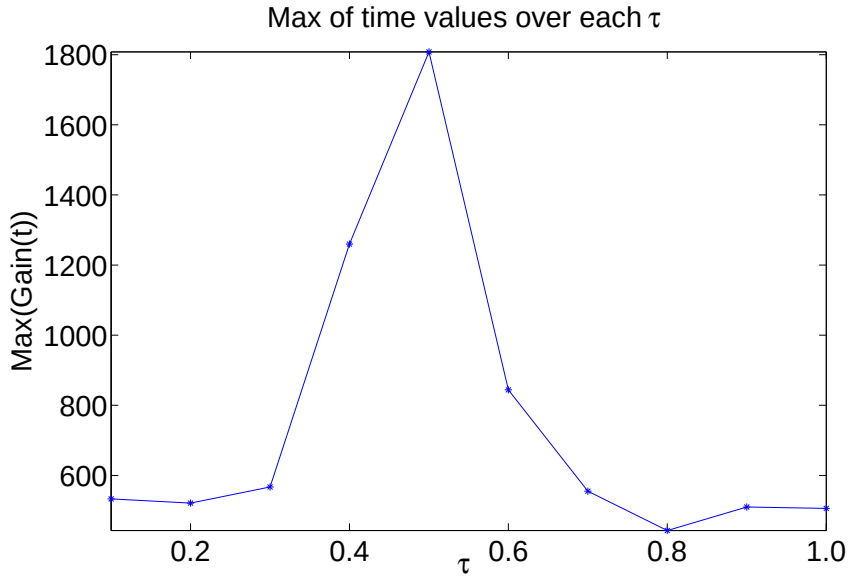


Figure 5.43. The maximum gain values of each τ over time or the sensor arrangement seen in Fig. 5.40.

between the 0.4 and 0.5 seconds. In this plot, the fluctuation between delays of the plume is even more apparent, and further investigation should be conducted to see whether it has a relation to the modulation rate of the plume. Finally, the conversion from delay to angle is computed. It is determined that a finer interpolation is needed if each angle degree is to

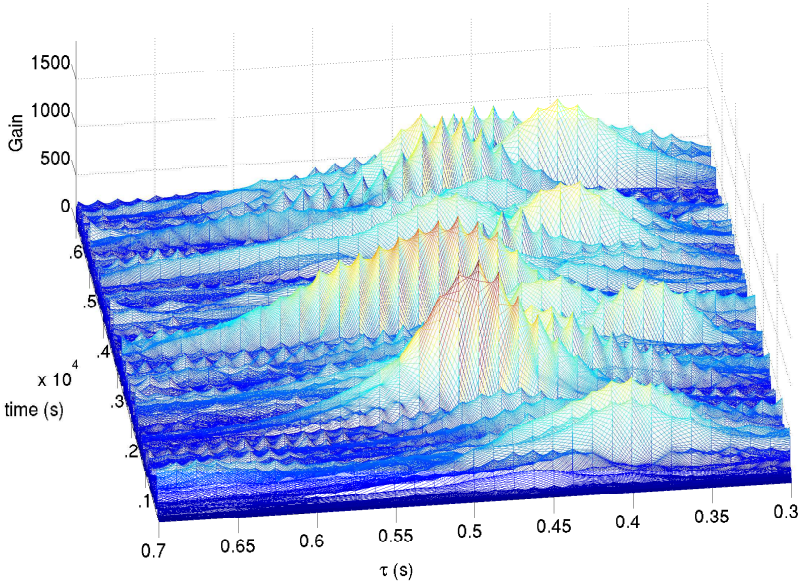


Figure 5.44. A plot of the gain, $y(t)$, vs the delay vs. time with an interpolation of the time data by ten for the sensor arrangement seen in Fig. 5.40. τ is sampled at $1/80$ of a second. It is shown that values of τ can be anywhere from 0.35 to 0.55 seconds, perhaps indicating the fluctating nature of the modulation in the plume.

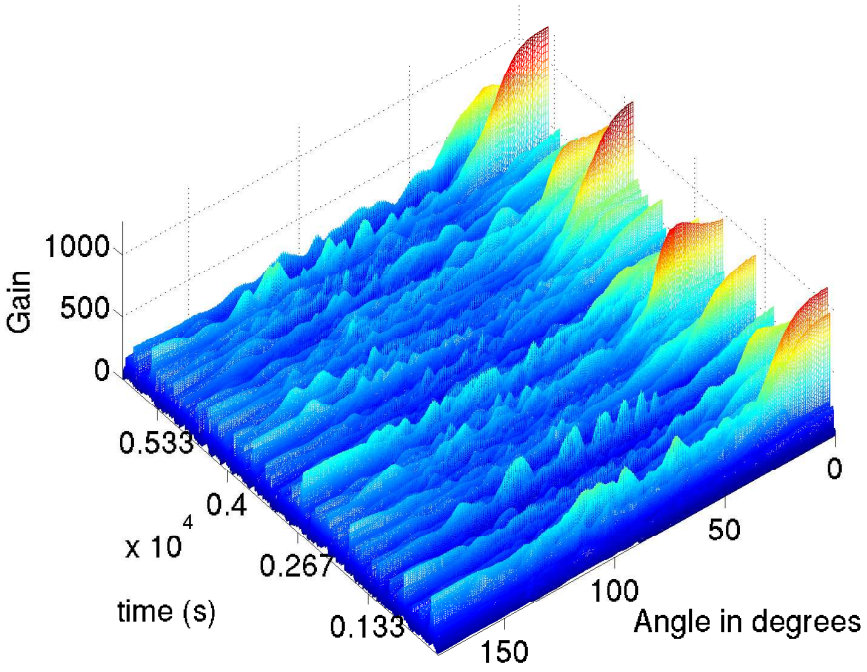


Figure 5.45. A plot of the gain, $y(t)$, vs the angle vs. time with an interpolation of the time data by fifteen for the sensor arrangement seen in Fig. 5.40. Each angle degree is tested.

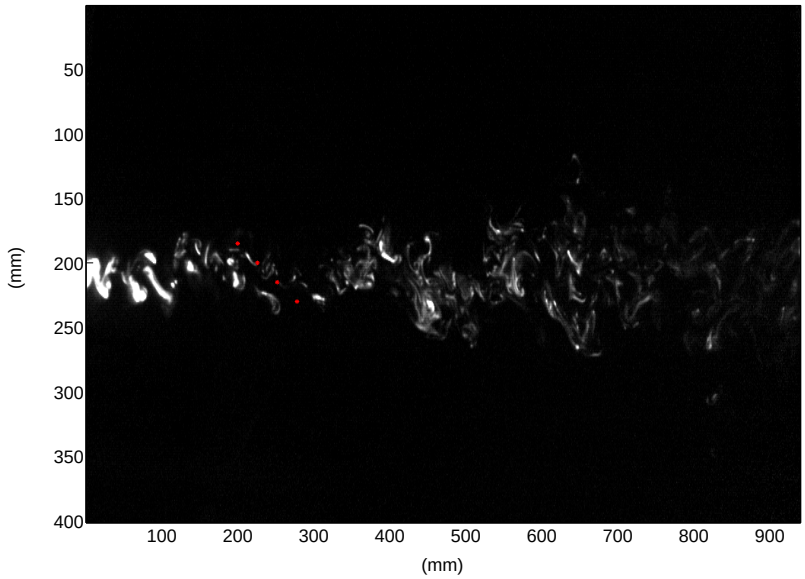


Figure 5.46. A diagonal linear array with 4 sensors, with 3 cm (30 pixel) spacing, an angle of -30° which makes the sensor separation at a distance of approximately 25 mm apart on the x-axis, starting at 20 cm down the plume.

be tested. In order to have a unique sample for every delay from the $sin(\theta)$ function, the following equation is used to determine the interpolation factor needed:

$$interp\_rate = ceil(\frac{1}{\tau}) \text{ where } \tau = \frac{d}{v} * sin(1^\circ) * cos(angle^\circ of array)$$

where d is the pixel distance between sensors, v is in pixels per time sample, and $angle^\circ of array$ is the angle orientation in degrees of the array. For the Gain vs. Angle vs. Time graph seen in Fig. 5.45, the interpolation rate is 15 (20/5*0.175*1). In this graph, we can see that the delays correspond to a gain in approximately the 0° direction. This is the direction of the flow with respect to the horizontal array and correspondingly, the AOA is 180°. But due to the delay variability seen in Gain vs. Delay vs. Time graph, the gain along the angles seem to be strong from 0° to 20°, again emphasizing that there might be a fluctuation in angle the plume is hitting the array as well, and this phenomenon should be further investigated.

Other array placements are examined. An array of interest is a diagonal array since it will capture vertical as well as horizontal time delay in the plume. This array is placed directly in the plume seen in Fig. 5.46, thus it is limited to 4 sensors if the sensor distances

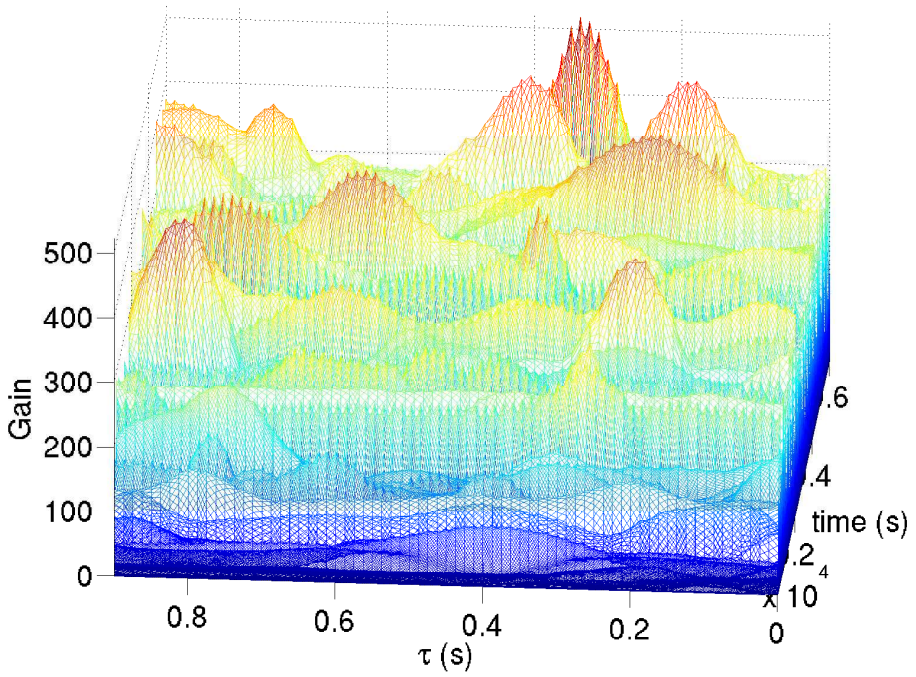


Figure 5.47. A plot of the gain, $y(t)$, vs the delay vs. time with an interpolation of the time data by ten for the sensor arrangement seen in Fig. 5.46. τ is sampled at $1/80$ of a second over values 0 to 9.

are to be comparable to the previous arrangement. A 30 mm sensor separation is used, which corresponds to about a $30cos(30°) = 26$ mm/pixel spacing on the x-axis. If the plume is travelling at 0° with respect to this −30° array, then the expected τ is $26/50 = 0.52$ seconds. The corresponding Gain vs. Delay vs. Time graph is seen in Fig. 5.47. While many ”bumps” occur, none of them line up to a particular delay, and in fact, there are no notable gains at the expected 0.52 second delay. The corresponding Gain vs. Angle vs. Time graph is seen in Fig. 5.48. Interestingly enough, in the angle plot, the areas of high gain seem to be clustered around 0°, 60°, 140°, and 180°. Although, the location of a 60° angle in relation to the −30° angle array (overall 30°) does not make intuitive sense.

In Fig. 5.49, one more orientation was tested for a diagonal 10-sensor array but further down the plume. Running the delay-and-sum beamformer, we obtain Fig. 5.50. In these results, there is no clear τ which there is a gain. The farther out from the plume we get, the lower the amplitude of the parcels and the more dispersed they are, so this may attribute to

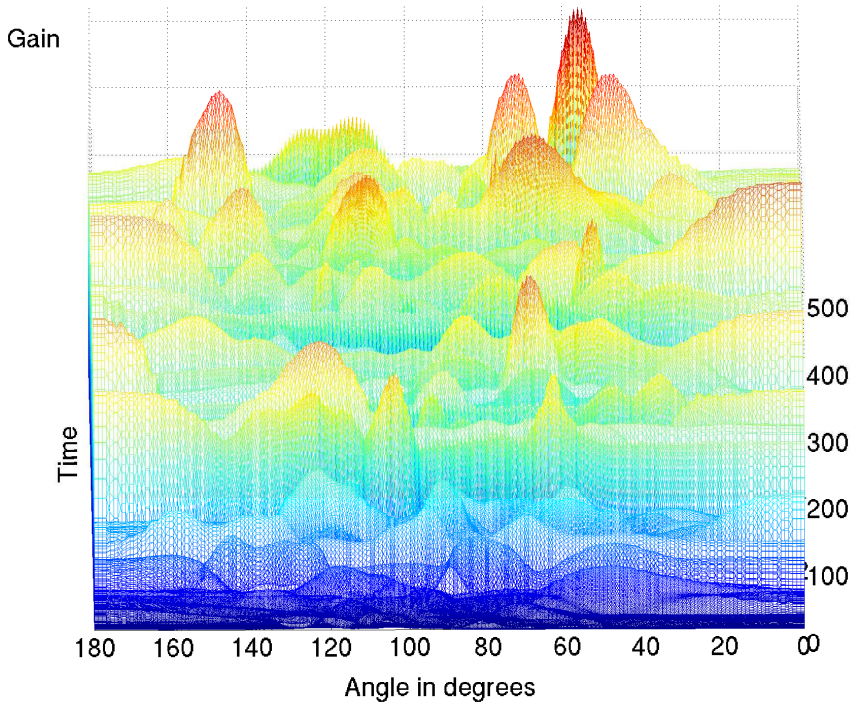


Figure 5.48. A plot of the gain, $y(t)$, vs the angle vs. time with an interpolation of the time data by twelve for the sensor arrangement seen in Fig. 5.46. Each angle degree is tested.

insignificant gains in the plot. The one notable gain seems to be at τ= 0.3 seconds which has little physical meaning. This sensor arrangement also corresponds to little significant gains in Gain vs. Angle vs. Time plot. If anything, there seems to be a consistent peak (at least for 3 instances) that is around 120°, but this does not intuitively make sense, as it would correspond to the plume orthogonal to the direction it is currently coming from.

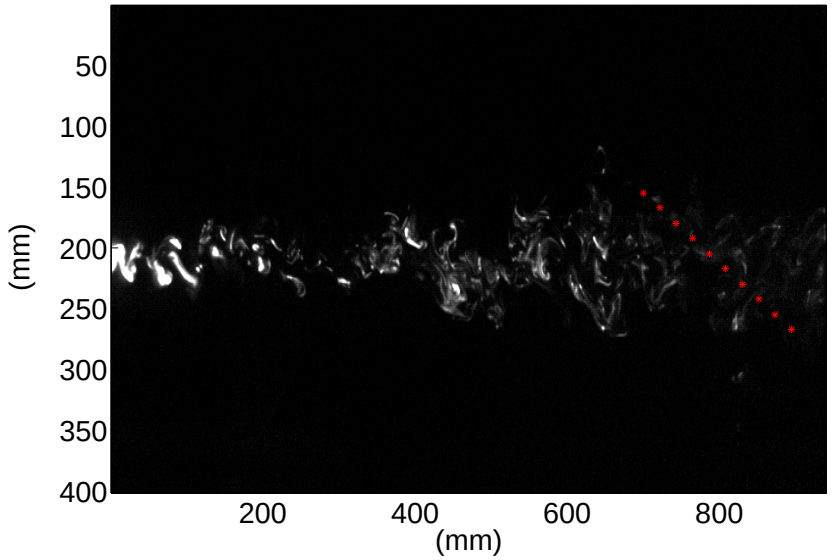


Figure 5.49. A diagonal linear array with 10 sensors, with 2.5 cm (25 pixel) spacing, an angle of $-30°$ which makes the sensor separation approximately 25 mm apart on the x-axis, starting at 70 cm down the plume.

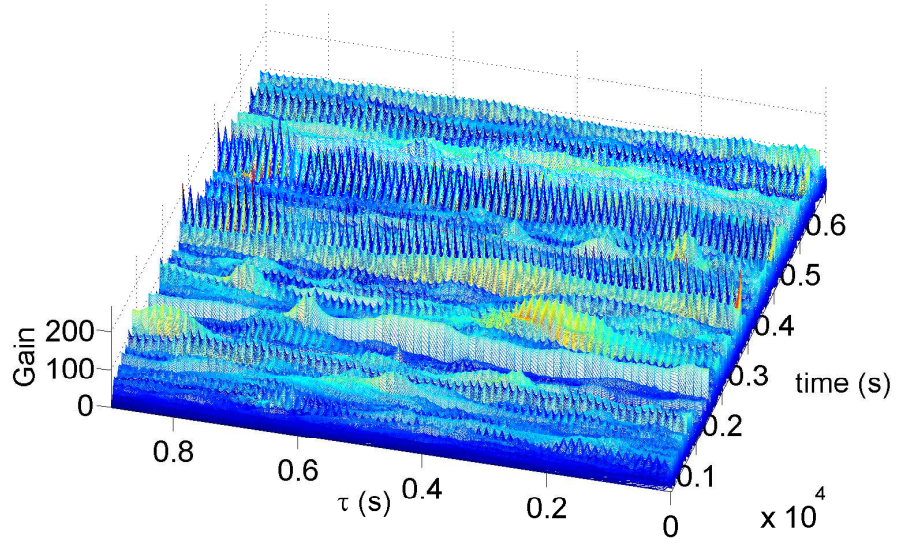


Figure 5.50. A plot of the gain, $y(t)$, vs the delay vs. time with an interpolation of the time data by ten for the sensor arrangement seen in Fig. 5.49. τ is sampled at $1/80$ of a second over values 0 to 9.

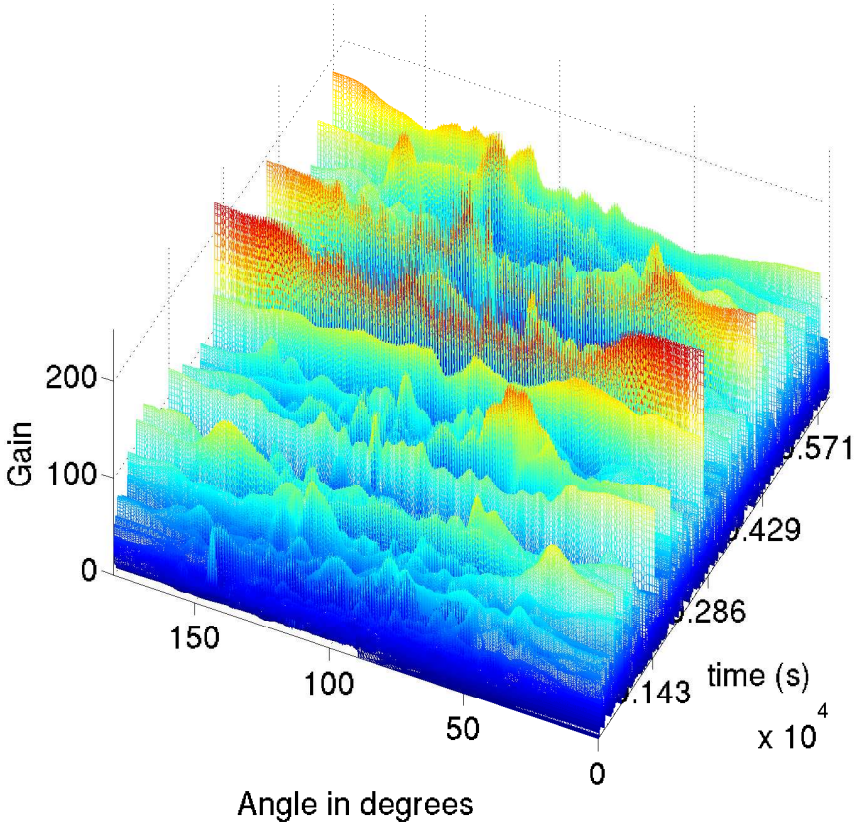


Figure 5.51. A plot of the gain, $y(t)$, vs the angle vs. time with an interpolation of the time data by fourteen for the sensor arrangement seen in Fig. 5.49. Each angle degree is tested.

CHAPTER 6

CONCLUSIONS

The central theme of this research explores the intersection between biology and signal processing. We show how signal processing techniques can be used to reverse-engineer biology, such DNA sturcutre, and also how signal processing can be used to extend and refine biological processes into practical applications on the functional level.

First, we develop and show how linear algebraic techniques can be used to analyze DNA. When these linear techniques are used for a strict conditions, that of universal error-correction in the sequence, they are not fool-proof, but more general linear methods are superior in detecting imperfect periodicities (approximate tandem repeats), a classically difficult problem.

Second, we show that an engineered system, in this case, odor localization, can be improved via signal processing and biological techniques. We show that chemoreceptor clustering improves sensor array performance, and there is potential for improvement as the chemotaxis model complexity increases.

Third, we analyze turbulent plume experimental data and show that a cross-correlation method, such as interaural time delay in binaural hearing, improves turbulent plume localization. In addition, the overall research supplies concrete examples of how signal processing techniques can be used to analyze biology and how biology can help us engineer better systems.

6.1 Impact of Thesis

This research contributes to our understanding of bio-informatics as well as improves practical chemical localization.

Contributions of the thesis include the following:

- The problem of assigning arbitrary numerical values to nucleotides is discussed. We

propose application of a finite-field mathematical framework for symbolic DNA analysis. Despite the specific mapping, the numerical nucleotide values can still maintain a symbolic nature. [81]

- Linear algebraic methods are developed for detecting a general error-correcting code in DNA. While the subspace partitioning test did not find a consistent linear block code in DNA, it is nonetheless an interesting technique to look for coding structure in other signals. Also, regions are found to have some coding structure which may indicate regions of preservation. [86]

- A tandem and approximate tandem repeat detection algorithm is developed using a linear dependence test. Persistent rank-deficient frames are highlighted, and this is a much more general redundancy test than linear block coding. The algorithm detects repetitive regions even in high levels of DNA mutations. It was shown to detect a periodicity in a human satellite region that previous algorithms did not. [85]

- Previous chemotaxis-inspired techniques have mainly focused on random walk strategies or gradient following. In this thesis, a multi-sensor yet single-node algorithm is developed using inspiration from chemoreceptor cooperation. The chemoreceptor signaling is implemented with Hebbian learning, exhibited in other types of biological learning such as neurons. Various geometric interpretations of the connections between receptors were evaluated and the localized clustering seen in nature, performed the best. [82] [83]

- The chemoreceptor cooperation algorithm was implemented for a stationary temperature sensor array and tested in several environmental scenarios. It is shown that for the stationary case, there is a trade-off between precision of angle localization and variance of the convergence [84].

- By testing the algorithm in several various environmental scenarios, it became apparent that turbulence needs more investigation. Experimental Kármán Vortex Street plume data was obtained and evaluated for frequency content through single-sensor fourier analysis and multiple sensor correlation and coherence. It was discovered that it is difficult to gain much information out of the plume without spectral phase information.

- It was found the spectral phase information of small events, or parcels, in the plume can be exploited. A pairwise cross-correlation method is developed to determine wind angle-of-arrival for turbulent plume localization. [80] A Delay-and-sum beamforming method is also investigated.

6.2 Future Work

- A finite-field framework of $GF(4)$ was used for DNA analysis, but a finite-field framework should be extended to amino acid sequence analysis. This could aid in protein structure prediction and periodicity analysis in these sequences.

- Also, we specifically looked for an error-correcting block code in DNA. It has been conjectured that DNA is a nested process, thus a convolutional coder would be a more suitable model. Finding a convolutional coding model with unknown parameters is a challenging problem.

- Our tracking methods are all two-dimensional. Expanding the solutions to 3-D is of practical importance.

- Mathematical models of chemotaxis and chemoreceptor clustering are being developed. Simulating these models are essential to understanding the process.

- Also, a broad area for further analysis is to look into other olfaction mechanisms (e.g. mammalian) which can help odor and chemical tracking.

- The experimental plume data showed that the effect of diffusion was still governing the plume. It may be feasible to time-average the sensor data to filter out short-time effects and retrieve the diffusive field. Then an algorithm designed for diffusive navigation (e.g. our chemoreceptor-inspired algorithm) can be used.

- Implementing chemical localization algorithms in lower-power analog electronics. A new technology, Floating-Gate Field-Programmable Analog Arrays (FG-FPAAs) [37], enable rapid prototyping of complex analog systems. FPAA's can make DSP algorithms possible in lower-power analog circuitry. Implementation of such a chemical localization systems have unlimited applications for security, space exploration, and disease detection.

APPENDIX A

PROPAGATING WAVES, DIFFUSION, AND ARRAY SIGNAL PROCESSING

Array beamforming is a technique in which an array of sensors is exploited to achieve maximum reception in a specified direction (in the presence of noise) while other signals are rejected [48].

The propagating signals of interest may be transverse electromagnetic waves and compressional acoustic waves in various mediums. These waves can be modeled with the wave equation in 3-D derived from Maxwell's equations:

$$\frac{\partial^2 s}{\partial x^2} + \frac{\partial^2 s}{\partial y^2} + \frac{\partial^2 s}{\partial z^2} = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2} \quad \text{(cartesian coordinates)}$$

$$\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial s}{\partial r}\right) = \frac{1}{c^2}\frac{\partial^2 s}{\partial t^2} \quad \text{(spherical coordinates)}$$

Most of array signal processing techniques rely on the monochromatic, planar wave solution meaning that the signal is periodic and has constant wavefronts, which are planes of constant phase. This solution to the monochromatic, planar wave, as seen in Fig. A.1, is:

$$s(\vec{x}, t) = Ae^{j(\omega t - \vec{k}\cdot x)}$$

where $\vec{k}$ is the wavenumber vector (related to the direction of propagation), ω is the frequency in radians, t is time, and $\vec{x}$ is spatial position.

The solution to the wave equation for a spherical wave, shown in Fig. A.2, is:

$$s(r, t) = \frac{A}{r}e^{j(\omega t - kr)}$$

The physcial laws associated with chemicals and heat are however different; they obey

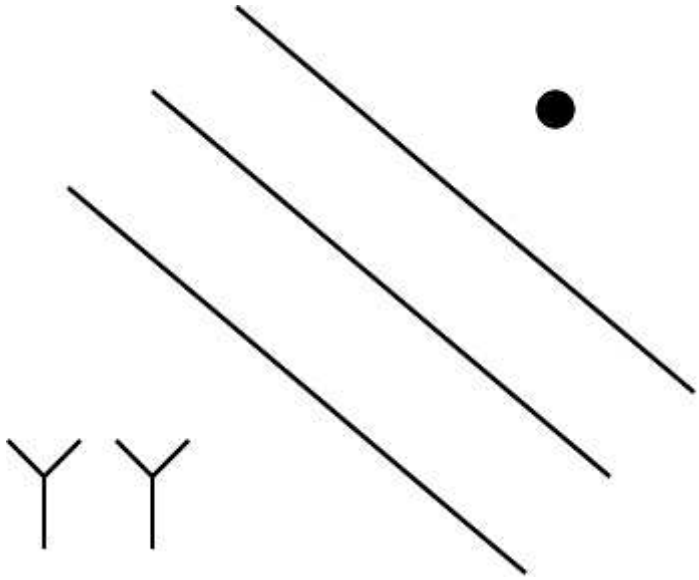


Figure A.1. A planar wave propagation.

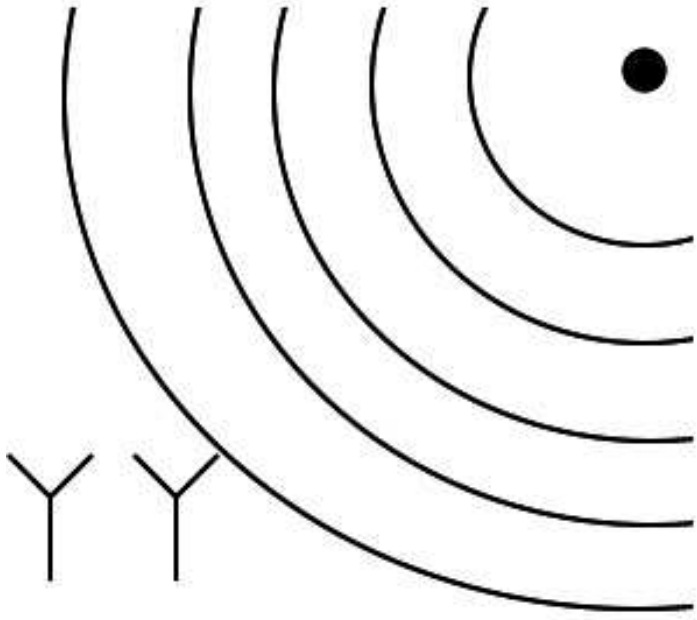


Figure A.2. A spherical wave propagation.

diffusive laws. Fick's first law of diffusion states:

$$J = -D\frac{\partial C}{\partial x}$$

where J is the diffusion flux, D is the diffusion coefficient, C is the concentration, and x is the position. This equation is used in steady-state diffusion, where the concentration in the diffusion volume does not change with respect to time.

When it does change with respect to time, Fick's second law of diffusion states:

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}$$

In order to solve Fick's second law, one initial condition and two boundary conditions are required. For a one-dimensional, continuous point release in 2-D with no boundaries where $C(r, 0) = 0$, $C(0, t) = \mu$, and $C(\infty, t) = 0$, the solution is:

$$C(r, t) = \frac{\mu}{4\pi Dr}\text{erfc}(\frac{r}{2\sqrt{Dt}})$$

For an instantaneous release: $C(r, 0) = 0$, $\int C(r, t)dr = \mu$, and $C(r, \infty) = 0$, the solution is:

$$C(r, t) = \frac{\mu}{4\pi Dt^{3/2}}e^{-\frac{r^2}{4Dt}}$$

The most important assumption for array processing is that the signal be a transverse wave with space and time linked by $t - \vec{\alpha} \cdot \vec{x}$ in the propagating wave, $s(\cdot)$. where $\vec{\alpha}$ is the slowness vector. This means we can reconstruct the signal by temporally sampling or spatially sampling the signal.

This leads us to *spatiotemporal filtering*. Sensor outputs can be delayed by appropriate amounts and added together to reinforce the signal with respect to noise or waves in different directions. Weights can also be used to amplify or attenuate the signal and are called *shadowing*. The output of the such a selective constructive/deconstructive adder, or delay-and-sum beamformer is:

Fig. A.3 illustrates a classical linear array beamformer that computes:

$$x(t) = \mathbf{a}^T(\theta)\mathbf{s}(t) \tag{A.1}$$

where $\mathbf{s}(t)$ is a vector of sensors inputs, $[s_1(t), s_2(t), \ldots s_n(t)]^T$. $\mathbf{x}(t)$ is the beamformer output, $[x_1(t), x_2(t), \ldots x_n(t)]^T$, and $\mathbf{a}(\theta) = [a(\theta_1), a(\theta_2), \ldots a(\theta_n)]^T$ is the steering vector looking in signal wavefront direction, θ. It is important to note that $s_k(t)$ has the following relationship to a sensor point in the field:

$$s_k(t) = a_k(\theta_k)s(t - \tau_k)$$

where τ_k is the time delay from the reference point and $a(\theta_k)$ is the steering vector for direction, θ_k. Sensor noise can be introduced to the measurements:

$$\mathbf{y}(t) = \mathbf{s}(t) + \mathbf{n}(t)$$

and $\mathbf{y}(t)$ can be substituted into $\mathbf{s}(t)$ in (A.1) to obtain:

$$x(t) = \mathbf{a}^T(\theta)\mathbf{s}(t) + n(t) \tag{A.2}$$

The signal can be amplified or attenuated by adjusting the beamformer weights, $\mathbf{a}(\theta)$ or the time delays, τ. For a linear array in a planar wavefield, the time delay is related to the direction-of-arrival angle of a wavefront:

$$\tau_k = \frac{d_k}{c}\sin(\theta_k) \tag{A.3}$$

where d_k is the distance between each sensor

For nonwave fields such as concentration gradients that obey diffusive laws, there is no phase present in the field, and (A.3) does not make sense. Therefore, θ is not used in the nonwave field model and we are left with the beamformer output:

$$x(t) = \mathbf{a}^T\mathbf{s}(t) + n(t) \tag{A.4}$$

where $\mathbf{a}^T = [\alpha_1, \alpha_2, \ldots \alpha_n]$, scalar shading constants.

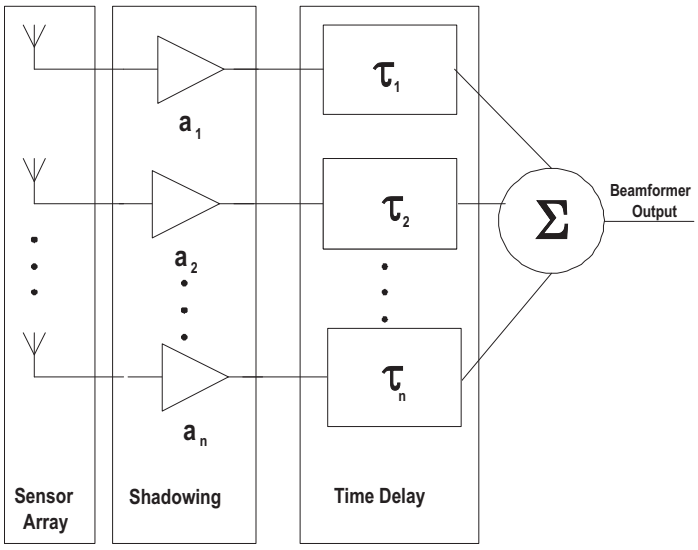


Figure A.3. Classical delay-and-sum beamformer.

APPENDIX B

HEBBIAN LEARNING AND LMS

Donald Hebb, a neurophysiologist, laid the foundation of neural computation. He verified that once a neuron repeatedly excited another neuro, the threshold of excitation in the latter neuron decreased. The excitation from the first neuron was thus amplified or the threshold to excite the second neuron was lowered. This is described by Hebb's rule [76]:

$$\triangle w_{ij} = \eta x_j y_i \tag{B.1}$$

where x_j is the pre-synaptic input and y_i is the post-synaptic output as seen in Fig. B.1.

$$w[n+1] = w[n] + \eta x[n] y[n] \tag{B.2}$$

There is no desired signal required in Hebbian learning thus making it a type of unsupervised learning. For the scalar case, when $y[n] = w[n]x[n]$, the Hebbian update is:

$$w[n+1] = w[n][1 + \eta x^2[n]] \tag{B.3}$$

If the initial value of the weight is positive, the update will always be positive and increase without bound through the iterations.

In the next stage, there can be a multiple-input synapse (see B.2), denoted in matrix notation as:

$$y = \mathbf{w}^T\mathbf{x} = \mathbf{x}^T\mathbf{w} \tag{B.4}$$

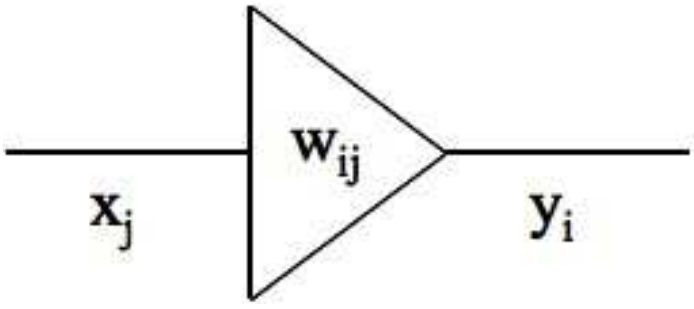


Figure B.1. Illustration of Hebb's Rule.

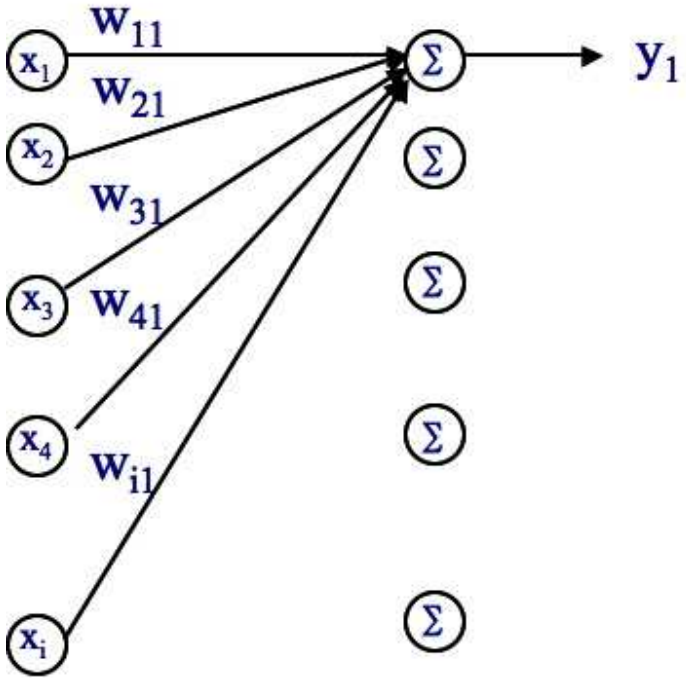


Figure B.2. A linear associator synapse.

which yields a correlated learning rule:

$$\begin{aligned} \triangle \mathbf{w} &= \eta y[n]\mathbf{x}[n] \\ &= \eta \mathbf{x}[n]\mathbf{x}^T[n]\mathbf{w}[n] \\ &= \eta \mathbf{R}_x \mathbf{w}[n] \end{aligned} \tag{B.5}$$

or in the update:

$$\mathbf{w}[n+1] = (\mathbf{I} + \eta \mathbf{R}_x)\mathbf{w}[n] \tag{B.6}$$

A matrix version of this update is:

$$\begin{aligned} \mathbf{W}[n+1] &= \mathbf{W}[n] + \eta \mathbf{R}_x \mathbf{W}[n] \\ \triangle \mathbf{W} &= \eta \mathbf{R}_x \mathbf{W} \end{aligned} \tag{B.7}$$

The update for each vector, $\mathbf{w}_i$ in $\mathbf{W}$ is discretized form of a first order linear system:

$$\frac{d\mathbf{w}_i}{dt} = \mathbf{R}_x \mathbf{w}_i \tag{B.8}$$

where $\mathbf{w}_i$ is each vector in $\mathbf{W}$. The solution to this first order linear system is $\mathbf{w}_i(t) = e^{\lambda t}\mathbf{u}$

where

$$\mathbf{R}_x\mathbf{U} = \Sigma\mathbf{U} \tag{B.9}$$

with $\mathbf{U}$ being the eigenvectors, $[\mathbf{u}_1\mathbf{u}_2 \ldots \mathbf{u}_n]$ of $\mathbf{R}$ and

$$\Sigma = \begin{pmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \lambda_n \end{pmatrix} \tag{B.10}$$

.

If we say that each $\mathbf{w}_i$ is a linear combinations of eigenvectors, $\mathbf{w}_i = \sum_j \alpha_j \mathbf{u}_j$ and substitute it into $\triangle\mathbf{w}_i = \eta\mathbf{R}_x\mathbf{w}_i$, we can say that the adaptation rate is proportional to the greatest eigenvalue, λ:

$$\triangle\mathbf{a}_i = \eta \sum_j \lambda_j \alpha_j \mathbf{u}_j \tag{B.11}$$

.

Unlike Hebbian learning, the least mean squares(LMS) adaptation results from a criterion minimizing the squared error function. Because of the error criterion, the "desired" signal that LMS is adapting to must be known, and is thus a form of supervised learning. Although this is a major difference between Hebbian correlated rule learning and LMS, they are similar learning methods.

The quadatric error function is defined:

$$\xi[n] = E|\mathbf{e}[n]|^2 \tag{B.12}$$

where $\mathbf{e}[n] = d[n] - \mathbf{w}^T[n]\mathbf{x}[n]$. $d[n]$ is the desired signal, $\mathbf{w}[n]$ is the weight vector (which can be vectorized to set of finite impulse response weights), and $\mathbf{x}[n]$ is the filter input.

The gradient, $\triangledown\xi[n]$, which is the tangent to the quadratic error will yield the direction towards the steepest ascent. An update equation to minimize the error using, $\mathbf{w}[n]$ is:

$$\mathbf{w}[n+1] = \mathbf{w}[n] - \mu \triangledown \xi[n] \tag{B.13}$$

The $\mathbf{w}[n]$ that minimizes the error function at time n is sought. The partial derivative with respect to the error equation is taken:

$$\frac{\partial \xi[n]}{\partial \mathbf{w}^*} = \frac{\partial E\{\mathbf{e}[n]\mathbf{e}^*[n]\}}{\partial \mathbf{w}^*} = \frac{E\{\mathbf{e}[n]\partial \mathbf{e}^*[n]\}}{\partial \mathbf{w}^*} \tag{B.14}$$

From here out, $\frac{\partial}{\partial \mathbf{w}^*}$ will be denoted as the gradient symbol ∇.

$$\nabla \mathbf{e}^*[n] = \nabla(d^*[n] - \mathbf{w}^{*T}[n]\mathbf{x}^*[n]) = -\mathbf{x}^*[n]$$

$$\nabla \xi[n] = -E\{\mathbf{e}[n]\mathbf{x}^*[n]\} \tag{B.15}$$

Therefore, substituting (B.15) into (B.13), the steepest descent equation becomes:

$$\mathbf{w}[n+1] = \mathbf{w}[n] + \mu E\{\mathbf{e}[n]\mathbf{x}^*[n]\} \tag{B.16}$$

The Hebbian learning weight update is similar to the weight error vector update in LMS. If $\mathbf{x}[n]$ and $d[n]$ are wide-sense stationary, $E\{\mathbf{e}[n]\mathbf{x}^*[n]\} = E\{d[n]\mathbf{x}^*[n]\} - E\{\mathbf{w}^T[n]\mathbf{x}[n]\mathbf{x}^*[n]\}$ $= \mathbf{r}_{dx}[n] - \mathbf{R}_x[n]\mathbf{w}[n] = 0$ so

$$\begin{aligned} \mathbf{w}[n+1] &= \mathbf{w}[n] + \mu(\mathbf{r}_{dx}[n] - \mathbf{R}_x[n]\mathbf{w}[n] \\ &= (\mathbf{I} - \mu \mathbf{R}_x[n])\mathbf{w}[n] + \mu \mathbf{r}_{dx}[n] \end{aligned} \tag{B.17}$$

Subtracting $\mathbf{w}$ from both sides and assuming $\mathbf{r}_{dx}[n] = \mathbf{R}_x[n]\mathbf{w}[n]$ when $\mathbf{x}[n]$ and $d[n]$ are WSS, the weight error vector is:

$$\mathbf{w}[n+1] - \mathbf{w} = (\mathbf{I} - \mu \mathbf{R}_x[n])\mathbf{w}[n] + \mu \mathbf{R}_x[n]\mathbf{w} - \mathbf{w} \tag{B.18}$$

Denoting $\mathbf{c}[n] = \mathbf{w}[n] - \mathbf{w}$, the weight error vector update is:

$$\mathbf{c}[n+1] = (\mathbf{I} - \mu \mathbf{R}_x[n])\mathbf{c}[n] \tag{B.19}$$

which is similar to the Hebbian correlated learning rule update except the input covariance matrix, $\mathbf{R}_x$, component is positive. So while the LMS weight error vector update is related

to the negative covariance matrix, the Hebbian update is related to the positive portion. This section has derived Hebbian learning and LMS methods to give the reader a comparison of the differences between supervised and unsupervised learning.

APPENDIX C

MATHEMATICS OF A RANDOM WALK

A random walk [56] is the sum of a Bernoulli process, I_n. I_n is an identical and independently distributed (i.i.d.) random process taking on values from set $0, 1$ with probability of p for 0 and $1 - p$ for 1. From probability, we know a bernoulli process has mean and variance of $E[I_n] = p$ and $VAR[I_n] = p(1 - p)$.

For a 1-D random walk, a particle changes position by $+i$ or $-i$ unit every time step. A Bernoulli random process can be defined as:

$$D_n = \begin{cases} i & I_n = 1 \\ -i & I_n = 0 \end{cases} \tag{C.1}$$

; this Bernoulli random process (or outcomes of a sequence of Bernoulli Random Variables) is illustrated in Fig. C.1. In terms of I_n, $D_n = i(2In1)$. Thus, $E[D_n] = 2i\ E[I_n] - i = 2ip - i$ and $VAR[D_n] = VAR[2iI_n - i] = 4i(2VAR[I_n]) = 4i(2p(1-p))$. Let S_n be the corresponding sum process (or random walk) of D_n. The mean and variance of S_n are respectively $nE[D_n]$ and $nVAR[D_n]$. The corresponding 1-D random walk to S_n, for $i = 1$, is illustrated in Fig. C.2.

Since these variables are independent, one can easily extend the random walk process to two dimensions with the x and y component having the 1-D random walk. A random walk can also be generated from uniformly distributed random integers, not just a Bernoulli RV. A 2-D random walk in Fig. C.3 was simulated with equi-probable integer step sizes, i, from -10 to 10.

Also, if on each step, the organism has an affinity towards the 45° angle due to higher concentration levels in that direction, and moves in this direction by $+1, +1(x, y)$ each step in addition to the random $i = \pm 10$, this 2-D random walk has a 10% bias shown in Figure C.4, compared to the 0% bias in Fig. C.3.

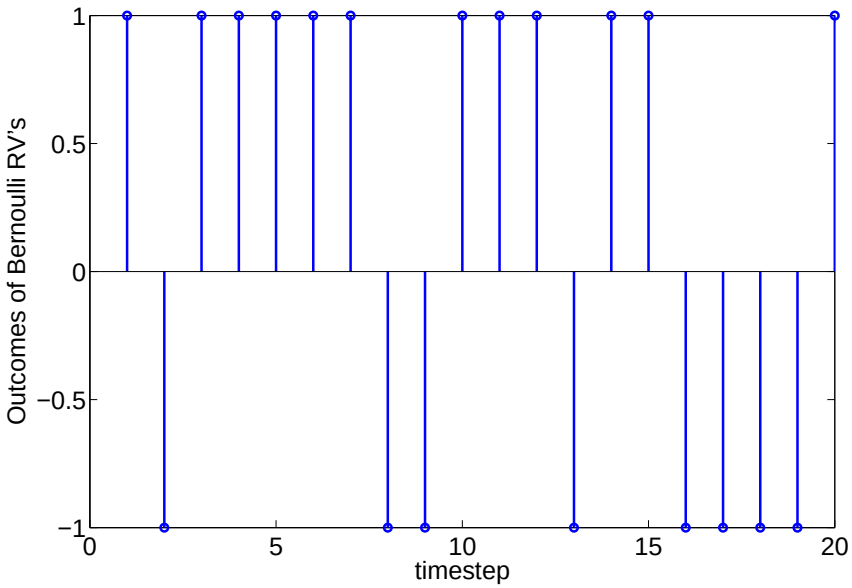


Figure C.1. Twenty outcomes of Bernoulli trials

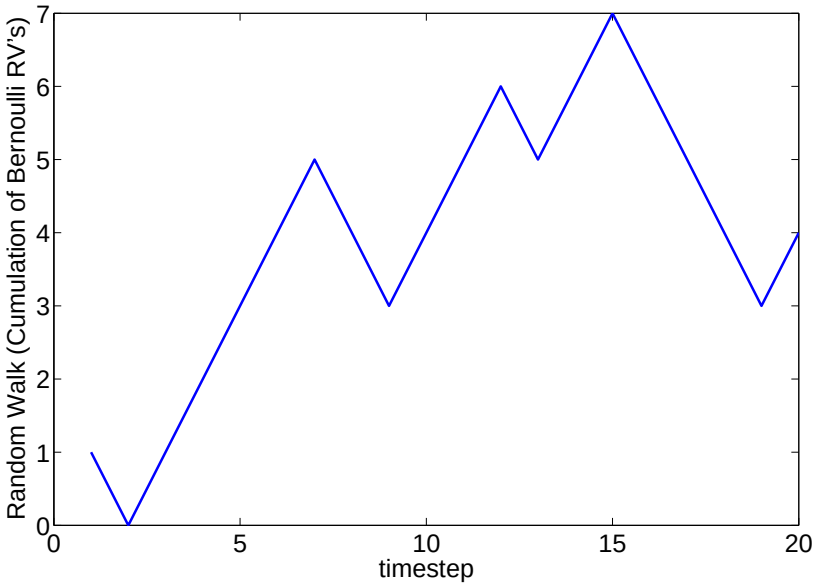


Figure C.2. Corresponding 1-D Random walk from the Bernoulli trials.

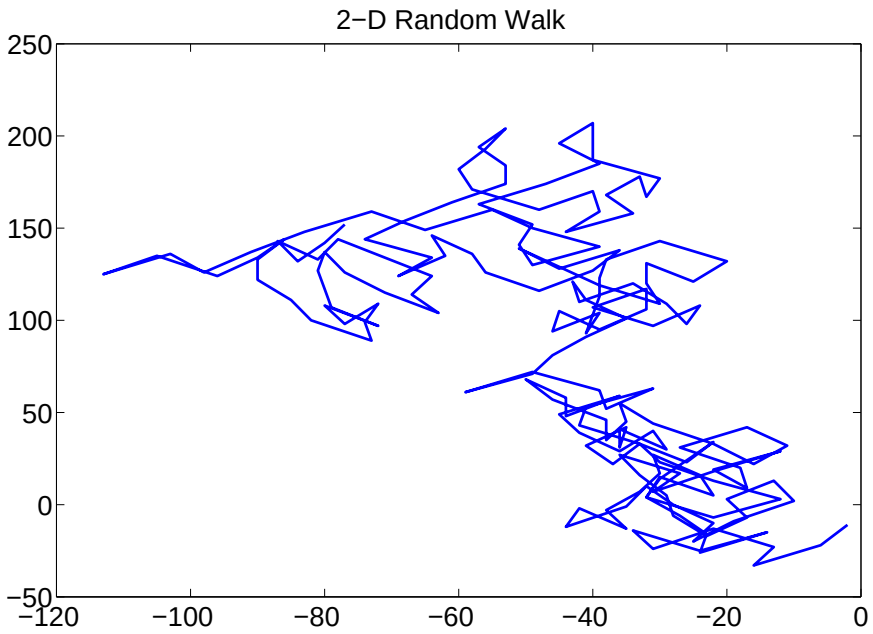


Figure C.3. 2-D Random walk from 200 length-10 steps.

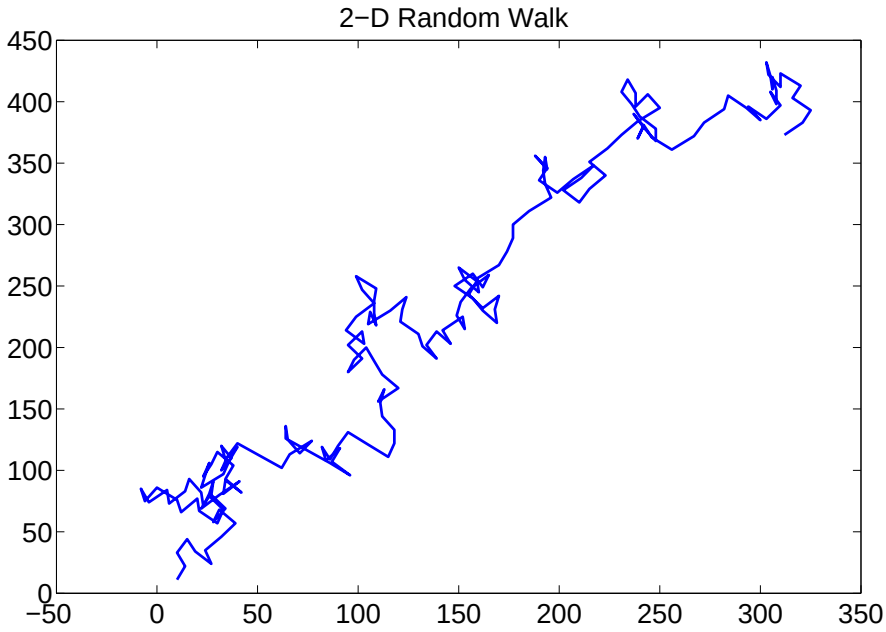


Figure C.4. 2-D Random walk with 10% bias from 200 steps.

REFERENCES

[1] "Incandescent lamps," in *Bulletin TP-110R1*, General Electric Co., 1980.

[2] *Brief History of Electronic Noses*. University of Tübingen, Chemistry Department, Weimar Research Group: http://www.ipc.uni-tuebingen.de/weimar/research/maintopics/multisensor/history_1.htm, Date accessed: July 2006.

[3] *GenBank: National Center for Biotechnology Database*. http://www.ncbi.nlm.nih.gov, Date accessed: July 2006.

[4] *Cyranose 320 Product*. Edgewood, MD: http://smithsdetection.com, Date accessed: June 2006.

[5] *Vapor Detection Technology*. Antwerp, Belgium: http://www.apopo.org, Date accessed: November 2005.

[6] *Reynolds number*. http://en.wikipedia.org/wiki/Reynolds_number: Wikipedia, Date accessed: September 2005.

[7] A , D., "Genomic signal processing," *IEEE Signal Processing Magazine*, July 2001.

[8] A , A., S , T., and H , P., "An adaptive front end for olfaction," in *IEEE International Symposium on Circuits and Systems*, vol. 3, (Lafayette, LA), pp. 107 – 110, June. 1998.

[9] A , D. and M , C. J., "A code in the protein coding genes," *Biosystems*, vol. 44, pp. 107–134, 1997.

[10] B , S., S , D., M., H., and K , I., "Targeting of the chemotaxis methylesterase/deamidase cheb to the polar receptor-kinase cluster in an escherichia coli cell," *Molecular Microbiology*, vol. 53.

[11] B , G., *An Introduction to Fluid Dynamics*. Cambridge, UK: Cambridge University Press, 1967.

[12] B , G., "Does information theory explain biological evolution?," *Europhysics Letters*, vol. 40, no. 3, pp. 343–348, 1997.

[13] B , G., "Replication decoding revisitied," in *IEEE Information Theory Workshop*, April 2003.

[14] B , D., H , B., and D W , S., "Analog vlsi circuits for odor discrimination," in *Proceedings of the 37th Midwest Symposium on Circuits and Systems*, vol. 2, (Lafayette, LA), pp. 1536 – 1539, Aug. 1994.

[15] B , P., L , A., and S , G. D., "Is there a code for protein-dna recognition? probab(ilistical)ly," *Bioessays (Functional Genomics and Bioinformatics)*, vol. 24, pp. 466–475, 2002.

[16] B , G., "Tandem repeat finder: a program to analyze dna sequences.," *Nucleic Acids Research*, vol. 27, pp. 573–580, 1999.

[17] B , H. C. and B , D. A., "Chemotaxis in escherichia coli analysed by three-dimensional tracking," *Nature*, vol. 239.

[18] B , M. and ., *GeneMark: A Family of Gene Prediction Programs*. http://opal.biology.gatech.edu/GeneMark/, Date accessed: July 2006.

[19] B , D., L , M. D., and M -F , C. J., "Receptor clustering as a cellular mechanism to control sensitivity," *Nature*, vol. 393, pp. 85–88, May 1998.

[20] B , M. and J , S., "Detection and visualization of tandem repeats in dna sequences," *IEEE Transactions on Signal Processing*, vol. 51, September 2003.

[21] B , R., *Genes and the Environment*. Taylor and Francis, Inc., Pennsylvania, 1999.

[22] C , K. and ., "Autoregressive modeling and feature analysis of dna sequences," *Eurasip Journal on Applied Signal Processing*, vol. 1, pp. 13–28, 2004.

[23] C , P., "Genomic signals of reoriented orfs," *Eurasip Journal on Applied Signal Processing*, vol. 1, pp. 132–137, May 2003.

[24] D , A., S , G. S., and R , A. A., "Bacterium-inspired robots for environmental monitoring," in *IEEE International Conference on Robotics and Automation*, (New Orleans, LA).

[25] D , C. and S , P., "A comparison between feature extraction methods of an electronic nose response," in *8th IEEE International Conference on Electronics, Circuits and Systems*, vol. 3, pp. 1243 – 1246, September 2001.

[26] D , D. B., "Spatial sensing of stimulus gradients can be superior to temporal sensing for free-swimming bacteria," *Biophysical Journal*, vol. 74, pp. 2272–2277, May 1998.

[27] F , A. M. and D , T. D., "Reactive localisation of an odour source by a learning mobile robot," in *Second Swedish Workshop on Autonomous Robotics*, (Stockholm, Sweden), October 2002.

[28] F , J., P , S., and L , W., "Plume mapping via hidden markov models," *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics*, 2003.

[29] F , D. and T , M., "The blueprint for life?," *IEEE Computer Magazine*, vol. 35, pp. 46–52, July 2002.

[30] F , J. P. and S , B., "Genomic engineering: Moving beyond dna sequence to function," *Proceedings of the IEEE*, vol. 88, no. 12, 2000.

[31] F , E. F., J., K. E., and C., B. E., "Integrating conflicting chemotactic signals: The role of memory in leukocyte navigation," *Journal of Cell Biology*, vol. 147.

[32] G , L., *Information Theory and the Living System*. Columbia University Press, New York, 1972.

[33] G , J. E. and ., "Evolutionary conservation of methyl-accepting chemotaxis protein location in bacteria and archaea," *Journal of Bacteriology*, vol. 182.

[34] G , G. and V L , C., *Matrix Computations*. The Johns Hopkins University Press, Maryland, 1996.

[35] G -O , R., "Pattern analysis for machine olfaction: a review," *IEEE Sensors Journal*, vol. 2, June 2002.

[36] H , W., "Telomerase and cancer: where and when?," *Clinical Cancer Research*, vol. 7, p. 29532954, October 2001.

[37] H , T., T , C. M., H , P., and A , D. V., "Application performance of elements in a floating-gate fpaa," in *IEEE Proceedings of the International Symposium on Circuits and Systems*.

[38] H , A., *Identification of Tandem Repeats Simple and Complex Pattern Structures in DNA*. PhD thesis, University of Madison, Wisconsin, 2002.

[39] H , A., *Beyond Tandem Repeats*. http://www.cs.wisc.edu/ gensoft/beyondTR/static/HSVDJSAT.html, Date accessed: July 2006.

[40] H , A., *Self-organized Robotic System Design and Autonomous Odor Localization*. PhD thesis, California Insitute of Technology, 2002.

[41] H , E., G , J., and L , E., "Electronic noses: A review of signal processing techniques," in *IEE Proc. Circuits, Devices and Systems*, vol. 146, pp. 297–310, Dec. 1999.

[42] H , Y. K., "Brief comments on junk dna: is it really junk?," *Complexity International*, vol. 9, pp. 1–12, 2002.

[43] H , S., K , L., and K , S., "Coding properties of dna languages," *Theoretical Computer Science*, vol. 290, pp. 1557–1579, 2003.

[44] I , H. and ., "Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors," in 1st *IEEE Conference on Sensors*, pp. 1665–1670, 2002.

[45] I , H., H , M., M , T., W , D. R., K , T., and J , J., "Spectrum analysis of chemical signals for navigation through turbulent plumes," *internal manuscript*, pp. 1–19, 2004.

[46] I , H., N , G., N , T., and M , T., "Controlling a gas/odor plume-tracking robot based on transient responses of gas sensors," *IEEE Sensors Journal*, vol. 5, no. 3, pp. 537–545, 2005.

[47] J ´, J. and N , A., "Landmine detection and localization using chemical sensor array processing," *IEEE Transactions on Signal Processing*, vol. 48, May 2000.

[48] J , D. and D , D. E., *Array Signal Processing*. Englewood Cliffs, NJ: Prentice Hall, 1993.

[49] K , E. and V , G., "Field theory based navigation for autonomous mobile machines," in *Intelligent Components for Vehicles Workshop*, (Portsmouth, UK), March 1998.

[50] K , T., J , P., I , H., and J , J., "Chemical plume tracking. 2. multiple-frequency modulation," *Analytical Chemistry*, vol. 73, pp. 3669–3673, 2001.

[51] K , R. and ., *mreps tandem repeat finder*. http://bioinfo.lifl.fr/mreps/, Date accessed: July 2006.

[52] K , E., K , M., and K , N., "Information decomposition method for analysis of symbolical sequences," *Physical Letters A*, vol. 312, pp. 198–210, 2003.

[53] L , G. M., "An algorithm for approximate tandem repeats.," *Journal of Computational Biology*, vol. 8, no. 1, pp. 1–18, 2001.

[54] L , D., B , O., and B , H., *CMOS Cantilever Sensor Systems: Atomic Force Microscopy and Gas Sensing Applications*. Springer.

[55] L , J., *Genetic Notes*. http://mason.gmu.edu/~jlawrey/biol471/geneticnotes.html, Date accessed: July 2004.

[56] L -G , A., *Probability and Random Processes for Electrical Engineering, 2nd ed.* Addison-Wesley.

[57] L , L., T , Y., T , A., and L , L., "Is there an error correcting code in the dna?," *Biophysical Journal*, vol. 71, pp. 1539–1544, 1996.

[58] L , A., R , D., and Z , A., "Gas source tracing with a mobile robot using an adapted moth strategy," in *Autonome Mobile Systeme (AMS), 18. Fachgespräch, Karlsruhe, 4. - 5. December* (, I., ed.), (Stuttgart, Germany), pp. 150–160, GDI, 2003.

[59] M D , D., "The role of error-coding in shaping the nucleotide alphabet: natures choice of a, u, c, and g," in *IEEE EMBS Intl .Conference special session on Communication Theory, Coding Theory and Molecular Biology,*, pp. 3850–3853, May 2003.

[60] M , J. R. and S , L., "Polar location of the chemoreceptor complex in the escherichia coli cell," *Science*, vol. 259.

[61] M J ., G. Q., S , M., and B , H. W. P., "Smartbadges: A wearable computer and communication system," in *CODES/CASHE '98*, (Seattle, Washington, U.S.A.), March 1998.

[62] M , L., N., A., and A , A. T., "Olfactory sensory system for odour-plume tracking and localization," in 2nd *IEEE Conference on Sensors*, (Toronto, Canada), pp. 418–423, 2003.

[63] M , E. E., "Towards a biological coding theory discipline," *New Thesis*, vol. 1, pp. 19–37, January 2004.

[64] M , E. E. and ., "Coding theory based models for protein translation initiation in prokaryotic organisms," in *Fifth International Workshop on Information Processing in Cells and Tissues (IPCAT)*, September 2003.

[65] M , E. E. and ., "An error-correcting code framework for genetic sequence analysis," *Journal of the Franklin Institute*, vol. 341, pp. 89–109, January-March 2004.

[66] M , E., *Analysis of Coding Theory Based Models for Initiating Protein Translation in Prokaryotic Organisms*. PhD thesis, North Carolina State University, 2002.

[67] M , D., J., B., and C., W., "Selection against frameshift mutations limits. microsatellite expansion in coding dna," *Genome Research*, vol. 10, pp. 72–80, 2000.

[68] M¨ , S., M , J., A , S., and K , P., "Optimization based on bacterial chemotaxis," *IEEE Transactions on Evolutionary Computation*, vol. 6, February 2002.

[69] N , H. and S , J. R., *Handbook of Machine Olfaction: Electronic Nose Technology*. Weinheim: Wiley-VCH, 2002.

[70] N , A., P , B., and P , E., "Detection and localization of vapor-emitting sources," *IEEE Transactions on Signal Processing*, vol. 43, January 1995.

[71] P , S., M , L., K , J., L , A., and I , P., "Responding to directional cues: A tale of two cells," *IEEE Control Systems Magazine*, vol. 24, pp. 77–90, August 2004.

[72] P , K. M., "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control Systems Magazine*, vol. 22, pp. 52–67, June 2002.

[73] P , B. J. and J , J., "Chemfet arrays for chemical sensing microsystems," in *IEEE International Sensors Conference*, (Orlando, FL).

[74] P , L., B , D. L., and V , M. A., "On finding convolutional code generators for translation initiation of escherichia coli k-12," in *IEEE EMBS Intl.Conference special session on Communication Theory, Coding Theory and Molecular Biology*, pp. 3854–3857, 2003.

[75] P , B. and N , A., "Localizing vapor-emitting sources by moving sensors," *IEEE Transactions on Signal Processing*, vol. 44, April 1996.

[76] P , J., E , N., and L , W., *Neural and Adaptive Systems: Fundamentals through Simulations*. New York, NY: John Wiley and Sons, Inc., 2000.

[77] R , B., G -G , A., P -L , A., and G -O , R., "Sensor-based machine olfaction with a neurodynamics model of the olfactory bulb," in *Proceedings of the Intelligent Robots and Systems Conference*, vol. 1, pp. 319–324, October 2004.

[78] R -R ' , R., B -G , P., and O , J. L., "Application of information theory to dna sequence analysis: a review," *Pattern Recognition*, vol. 29, no. 7, pp. 1187–1194, 1996.

[79] R , T. and W , D., "Biologically-inspired pattern recognition for odor detection," *Pattern Recognition Letters*, vol. 21, pp. 213–219, 2000.

[80] R , G. and H , P., "Chemical source localization in unknown turbulence using the cross-correlation method," in *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Toulouse, France), May 2006.

[81] R , G. L., "Examining coding structure and redundancy in dna," *IEEE Engineering in Medicine and Biology Magazine*, vol. Special Issue on Communication Theory, Coding Theory, and Molecular Biology, pp. 62–68, January/February 2006.

[82] R , G. L. and H , P. E., "Biologically-inspired odor localization using beamforming," in *IEEE Workshop on Genomic Signal Processing and Statistics*, (Baltimore, MD), May 2004.

[83] R , G. L., H , P. E., and S , M. T., "Modified hebbian learning implementation for localizing and tracking diffusive sources," *Internal Manuscript*, to be submitted.

[84] R , G. L., S , M. T., and H , P. E., "Circuit implementation of a 2-d gradient source localizer," in 3^{rd} *IEEE Conference on Sensors*, (Vienna, Austria), October 2004.

[85] R , G., "Finding near-periodic dna regions using a finite-field framework," in *IEEE Workshop on Genomic Signal Processing and Statistics)*, May 2004.

[86] R , G. and M , J., "Investigation of coding structure in dna," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, April 2003.

[87] S , D. C. and M , E. E., "Visualizing ecc properties of e. coli k-12 translation initiation sites," in *2nd IEEE Workshop on Genomic Signal Processing and Statistics*, May 2004.

[88] S , T. D. and ., "Information content of binding sites on nucleotide sequences," *J. Mol. Biol.*, vol. 188, pp. 415–431, 1986.

[89] S , T., "Some lessons for molecular biology from information theory," *Entropy Measures*, vol. 119, pp. 229–237, 2003.

[90] S , M. and B , C., "Compensation for nucleotide bias in a genome by representation as a discrete channel with noise," *Bioinformatics*, vol. 18, no. 4, 2002.

[91] S , G., G , J., C , M., G , K., and C , M., "System identification of electronic nose data from cyanobacteria experiements," *IEEE Sensors Journal*, vol. 2, June 2002.

[92] S , C., *Mathematical Theory of Genetics*. PhD thesis, Massachusettes Insitute of Technology, 1941.

[93] S , Y. N., L , Z., L , C., M , B. A., and K , E. C., "Charge-based chemical sensors: a neuromorphic approach by the chemoreceptive neuron mos transistors (cnmos)," *IEEE Trans. Electron Devices*.

[94] S , R. R. and ., "Triplet repeat dna structures and human genetic disease: dynamic mutations from dynamic dna," *Journal of Bioscience*, vol. 27, pp. 1–12, April 2002.

[95] S , V., "Receptor clustering and signal processing in e. coli chemotaxis," *Trends in Microbiology*, vol. 12, pp. 569–576, 2004.

[96] S , V. and B , H. C., "Localization of components of the chemotaxis machinery of esherichia coli using fluorescent protein fusions," *Molecular Microbiology*, vol. 37.

[97] S , N., "On circular coding properties of gene and protein sequences," *Croatia Chemica Acta*, vol. 4, no. 1, pp. 999–1008, 1999.

[98] S , P. and M , R., *Introduction to Spectral Analysis*. Prentice Hall, 1997.

[99] S , A., K , A., and A , D., "Spectrogram analysis of genomes," *Eurasip Journal on Applied Signal Processing*, vol. 1, pp. 29–42, 2004.

[100] T , S., "Visual study of unsteady separated flows around bodies," *Progr. Aerospace Science*, vol. 17, pp. 287–348, 1977.

[101] T , E. N., "3-, 10.5-, 200-, and 400-base periodicities in genome sequences," *Physica A*, vol. 249, pp. 511–516, 1998.

[102] V , P. P. and Y , B. J., "The role of signal-processing concepts in genomics and proteomics," *Journal of the Franklin Institute*, vol. 341, January-March 2004.

[103] V , S., L , Y., and W , T., "Maximum likelihood localization of a diffusive point source using binary observations," *Preprint*, 2005.

[104] V , N. and ., "A clustering method for repeat analysis in dna sequences.," *Genome Biology*, vol. 2, pp. 1–11, 2001.

[105] W , W. and J , D., "Linear transforms of symbolic data," in *IEEE Transactions on Signal Processing)*, vol. 10, pp. 628–634, March 2002.

[106] W , X. H. and ., "Review of application of coding theory in genetic sequence analysis," in *Healthcom Proceedings*, June 2003.

[107] W , M. J. and D , D. B., "Behavioral observations and computer simulations of blue crab movement to a chemical source in a controlled turbulent flow," *The Journal of Experimental Biology*, vol. 205, pp. 3387–3398, 2002.

[108] W , S., *Error Control Systems*. Prentice Hall, New Jersey, 1995.

[109] W , H. R., "Nucleosome structural features and intrinsic properites of the tataaacgcc repeat sequence," *The Journal of Biological Chemistry*, vol. 274, November 1999.

[110] W , B., *Telomerase*. http: //www.people.vcu.edu/~bwindle/Telomerase/telomerase.html, Date accessed: July 2006.

VITA

Gail Rosen spent her childhood in New Castle, Pennsylvania and then went to high school in Daytona Beach, Florida. She received a B.S. in Electrical Engineering with highest honors from the Georgia Institute of Technology in 1999. She subsequently received her M.S. and Ph.D. degrees from Georgia Tech in 2002 and 2006, respectively. For her graduate studies, she was a recipient of numerous prestigious awards, such as an NSF Graduate Research Fellowship, an NSF GK-12 Teaching Fellowship, and an AT&T Research Laboratories grant. She has also interned at AT&T Research Laboratories enhancing audio reconstruction playback in 2000 and interned at MIT Lincoln Laboratories in 2004 developing techniques to classify particle events in HVAC systems. She is a recipient of numerous awards: a Georgia Tech ECE outstanding teaching award in 2003, 2nd place in the IEEE Sensors best student paper competition in 2004, a 2005 SAIC Georgia Tech Outstanding Research Paper award, and a winner in the best student paper competition at the 2006 IEEE conference on signal processing (ICASSP). Her research interests are in DNA structure, mutation, and repair, and bio-inspired designs, especially for chemical source location. Other interests include audio signal processing, engineering education, synthesizer music, swimming, and travelling.